

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
(інститут)

Факультет інформаційних технологій
(факультет)

Кафедра Програмного забезпечення комп'ютерних систем
(повна назва)

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеня
магістра

(назва освітньо-кваліфікаційного рівня)

студента *Куракіної Анастасії Станіславівни*
(ПІБ)

академічної групи *121М-22-1*
(шифр)

спеціальності *121 Інженерія програмного забезпечення*
(код і назва спеціальності)

освітньої програми *«Інженерія програмного забезпечення»*
(назва освітньої програми)

на тему: *Розробка та дослідження ефективності впровадження веб-додатка оптимізації набору продуктів для збалансованого харчування*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи				
розділів:				
спеціальний	<i>Проф. Бердник М.Г.</i>			
Рецензент	<i>Проф. Гнатушенко В.В.</i>			
Нормоконтролер	<i>Проф. Лактіонов І.С.</i>			

Дніпро
2023

Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

ЗАТВЕРДЖЕНО:
Завідувач кафедри
Програмного забезпечення комп'ютерних
систем

(повна назва)
М.О. Алексєєв

(підпис) (прізвище, ініціали)
« » 20 23 Року

ЗАВДАННЯ
на виконання кваліфікаційної роботи магістра

спеціальності Інженерія програмного забезпечення
(код і назва спеціальності)

студенту 121м-22-1 Куракіної А.С.
(група) (прізвище та ініціали)

Тема кваліфікаційної роботи Розробка та дослідження ефективності
впровадження веб-додатка оптимізації набору продуктів для збалансованого харчування

1 ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Наказ ректора НТУ «Дніпровська політехніка» від 09.10.2023 р. № 1227

2 МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень – процес розробки та впровадження веб-додатку оптимізації набору продуктів для збалансованого харчування.

Предмет досліджень – математичні моделі та програмне забезпечення оптимізації продуктового кошику.

Мета роботи – удосконалення системи планування та вибору продуктів згідно рекомендацій здорового харчування.

Методи дослідження – базуються на принципах системного аналізу, функціонального аналізу, теорії баз даних. Також були використані методи оптимізації цільової функції з заданими обмеженнями.

3 ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна – представлені методи оптимізації процесу вибору продуктів задля підвищення ефективності організації здорового харчування.

Практична цінність полягає в тому, що розроблений програмний продукт можна буде використовувати у повсякденному житті для оптимізації витрат на продукти та контролю денної норми калорій та макронутрієнтів.

4 ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Результати досліджень мають бути подані у вигляді, що дозволяє побачити та оцінити безпосереднє використання запропонованих методів. В результаті роботи повинен бути розроблений веб-додаток на основі фреймворків Django та Angular, що працює у всіх сучасних браузерах.

5 ЕТАПИ ВИКОНАННЯ РОБІТ

Найменування етапів робіт	Строки виконання робіт (початок – кінець)
Аналіз теми та постановка задачі.	15.09.2023-29.09.2023
Дослідження характеристик видів дієтичного харчування та вивчення існуючих рішень	30.09.2023-20.10.2023
Розробка та дослідження ефективності впровадження веб-додатка оптимізації набору продуктів для збалансованого харчування	21.11.2023-08.12.2023

6 РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект від реалізації результатів роботи очікується позитивний завдяки пришвидшенню та автоматизації процесу планування бюджету та набору продуктів харчування.

Соціальний ефект від реалізації результатів роботи очікується позитивним, завдяки полегшенню процесу контролю за дієтичним харчуванням та балансом макронутрієнтів.

Завдання видав

(підпис)

Бердник М. Г.

(прізвище, ініціали)

Завдання прийняв до виконання

(підпис)

Куракіна А. С.

(прізвище, ініціали)

Дата видачі завдання: 15.09.2023 р.

Термін подання дипломного проекту до ЕК 12.12.2023

РЕФЕРАТ

Пояснювальна записка: 102 стр., 47 рис., 23 табл., 4 додатки, 41 джерело.

Об'єкт дослідження: процес розробки та впровадження веб-додатку оптимізації набору продуктів для збалансованого харчування.

Предмет досліджень: математичні моделі та програмне забезпечення оптимізації продуктового кошику.

Мета магістерської роботи: удосконалення системи планування та вибору продуктів згідно рекомендацій здорового харчування.

Методи дослідження: алгоритми та методи оптимізації функції з заданими обмеженнями.

Наукова новизна даної роботи знаходиться у представлених методах оптимізації процесу вибору продуктів задля підвищення ефективності організації здорового харчування

Практична цінність полягає у розробці програмного продукту, який можна буде використовувати у повсякденному житті для оптимізації витрат на продукти та контролю денної норми калорій та макронутрієнтів.

Область застосування. Розроблена інформаційна система може застосовуватися для вирішення повсякденних задач з планування раціону харчування, розрахунку бюджету на закупівлю продуктів та контролю за щоденним балансом білків, жирів та вуглеводів.

Значення роботи та висновки. Алгоритм оптимізації раціону харчування дозволяє контролювати та зменшувати витрати на продукти, підвищувати якість повсякденної дієти та спрощувати процес контролю за раціоном, що позитивно впливає на здоров'я людини, зменшує ризики виникнення низки хвороб, таких як цукровий діабет, хвороби серця.

Прогнози щодо розвитку досліджень. Покращити інформаційну систему додавши до розрахунку контроль балансу мікронутрієнтів та харчових волокон.

Список ключових слів: дієтичне харчування, макронутрієнти, веб-додаток, Django, Python, оптимізація, продуктовий кошик, сервер, SPA, Angular.

ABSTRACT

Explanatory note: 102 pages, 47 figures, 23 tables, 4 appendices, 41 sources.

Object of research: the process of development and implementation of a web application for optimizing a set of products for a balanced diet.

Subject of research is the mathematical models and software for shopping cart optimization.

Purpose of Master's thesis: improvement of the system of planning and selection of products according to the recommendations of healthy nutrition.

Research methods: algorithms and optimization methods for minimizing a function with given constraints.

Originality of research is in the presented methods of optimizing the process of product selection in order to increase the efficiency of the organization of healthy food.

Practical value of the results consists of the development of a software product that can be used in everyday life to optimize food costs and control the daily allowance of calories and macronutrients.

Scope of application. The developed information system can be used to solve everyday tasks of planning a diet, calculating the budget for the purchase of products and monitoring the daily balance of proteins, fats and carbohydrates.

The value of the work and conclusions. The diet optimization algorithm allows you to control and reduce food costs, increase the quality of your daily diet and simplify the process of diet control, which has a positive effect on human health and reduces the risk of a number of diseases, such as diabetes and heart disease.

Research forecast and development. Improve the information system by adding micronutrient and dietary fiber balance control to the calculation.

Keywords: Diet, Macronutrients, Web Application, Django, Python, Optimization, Shopping Cart, Server, SPA, Angular.

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ	12
1.1. Лікувальні та профілактичні дієти	12
1.2. Норми поживних речовин та обмеження продуктів при дієтичному харчуванні	14
1.3. Проблема оптимальної закупівлі продуктів відповідно до потрібного раціону харчування	19
1.4. Застосування автоматизації для плануванні продуктового кошика	21
1.5. Аналіз пов'язаних рішень	22
1.6. Дослідження вимог до програмної системи.....	24
1.7. Висновки	26
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ТА ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ	27
2.1. Підготовка вхідних даних	27
2.2. Дослідження математичної моделі.....	30
2.3. Вибір інструмента оптимізації набору продуктів.....	32
2.4. Мінімізація скалярної функції кількох аргументів при оптимізації набору продуктів	34
2.5. Висновки	34
РОЗДІЛ 3. СТВОРЕННЯ ТА АНАЛІЗ РОБОТИ ВЕБ-ДОДАТКА ОПТИМІЗАЦІЇ НАБОРУ ПРОДУКТІВ.....	36
3.1. Архітектура веб-додатку оптимізації набору продуктів для збалансованого харчування.....	36
3.1.1. Опис структури сервера	43
3.1.2. Опис структури клієнта.....	45

	7
3.1.3. Інструменти розробки та розгортання додатку	47
3.1.4. Технічні вимоги до розгортання та запуску сервісів	48
3.2. Структура та опис бази даних.....	48
3.3. Розробка API інтерфейсу сервера.....	55
3.4. Опис програмного продукту	59
3.5. Аналіз ефективності моделювання набору продуктів	72
3.6. Висновки	79
ВИСНОВКИ.....	81
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	83
ДОДАТОК А. КОД ПРОГРАМИ.....	89
ДОДАТОК Б. ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ	96
ДОДАТОК В. ТЕЗИ ДОПОВІДІ НА МІЖДУНАРОДНІЙ КОНФЕРЕНЦІЇ.....	97
ДОДАТОК Г. СЕРТИФІКАТ УЧАСНИКА МІЖНАРОДНОЇ КОНФЕРЕНЦІЇ.....	102

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

BMR – Basal metabolic rate;

ІМТ – індекс маси тіла;

SPA – Single Page Application;

REST – Representational State Transfer;

UML – Unified Modeling Language;

ОС – операційна система;

БД – база даних;

PIP – Python Package Index;

ІМТ – індекс маси тіла;

SLSQP – Sequential Least Squares Programming

API – Application Programming Interface;

ORM – Object-Relational Mapping;

JWT – JSON Web Token.

ВСТУП

Вплив харчування на здоров'я підтверджено не одним поколінням вчених, лікарів, нутріціологів. Споживання основних білків, жирів та вуглеводів, вітамінів та харчових волокон, їх співвідношення та кількість суттєво впливають на функціонування організму, роботу його внутрішніх процесів.

Актуальність дослідження полягає у необхідності надання зручного інструменту планування продуктів харчування для збалансованої дієти, контролю бюджету на їх закупівлю, а також оптимізація та пришвидшення цих процесів.

Вибір правильного раціону харчування залежить від цілей та стану організму. Так багато захворювань, наприклад, цукровий діабет, артеріальна гіпертензія, хвороби серця, для свого лікування та профілактики потребують дотримання відповідних дієт.

Через це розуміння рекомендацій та обмежень дієти є важливим аспектом здорового харчування. Дослідження цих особливостей, їх оформлення для зручного перегляду та маніпуляцій, а також надання швидких рекомендацій щодо закупівель дозволить підвищити зручність та ефективності контролю за здоров'ям.

У реаліях нестабільної економічної ситуації обмеження бюджету може стати наріжним каменем організації збалансованого харчування, адже постає питання ефективного вибору продуктів, так щоб з найменшими тратами, отримувати найбільшу кількість енергії та корисних речовин у потрібному співвідношенні.

Вивчення та аналіз предметної галузі дозволили зробити висновок про те, що хоча існує велика кількість додатків для розрахунку калорій та підбору здорового раціону харчування, більшість з них не враховує реалії українського ринку та можливості поєднувати планування продуктового кошику з контролем його вартості.

Мета дослідження полягає в розробці та дослідження ефективного впровадження веб-додатку оптимізації набору продуктів для збалансованого харчування.

Об'єкт дослідження: процес розробки та впровадження веб-додатку оптимізації набору продуктів для збалансованого харчування.

Предмет дослідження: математичні моделі та програмне забезпечення оптимізації продуктового кошику.

Методи дослідження: базуються на принципах системного аналізу, функціонального аналізу, теорії баз даних. Також були використані методи оптимізації цільової функції з заданими обмеженнями.

Наукова новизна: представлені методи оптимізації процесу вибору продуктів задля підвищення ефективності організації здорового харчування.

Практична цінність полягає в тому, що розроблений програмний продукт можна буде використовувати у повсякденному житті для оптимізації витрат на продукти та контролю денної норми калорій та макронутрієнтів.

Особистий внесок автора:

1. Розробка теоретичної частини роботи, де було досліджено та систематизовано знання щодо дієтичного харчування, балансу макронутрієнтів рекомендованих для різних типів дієт;
2. Розробка алгоритму підготовки вхідних даних та побудова математичної моделі.
3. Вибір методів досліджень і технологій реалізацій;
4. Створення веб-додатку для оптимізації продуктового кошику. Формування основної бази даних продуктів харчування для обчислень, їх актуальної вартості та калорійності;
5. Аналіз та порівняння отриманих результатів.
6. Оцінка отриманих результатів.

Структура і обсяг роботи. Робота складається з вступу, трьох розділів, висновків, списку використаних джерел і двох додатків. Загальний обсяг роботи становить 102 сторінок, із них 69 сторінок основного тексту, в тому

числі 47 рисунків, 26 таблиць і список використаних джерел із 41 найменувань на 6 сторінках.

РОЗДІЛ 1

АНАЛІЗ ТЕМИ ТА ПОСТАНОВКА ЗАДАЧІ

1.1. Лікувальні та профілактичні дієти

Дієтологія — практична наука, що вивчає та описує принципи здорового харчування. Вплив їжі на людину досліджували протягом довгого часу. Від Гіппократа, що вчив лікарів знатися на властивостях поживних продуктів, рекомендацій римського лікаря Асклепіада, де описувалися особливості споживання їжі та напоїв при різних захворюваннях, до американського вченого Вілбура Оліна Атватера, що став батьком сучасної освіти в галузі харчування, досліджував особливості метаболізму організму людини, розкрив енергетичну цінність продуктів та виділив основні харчові макронутрієнти: білки, вуглеводи та жири [1-2]. Він став одним з основоположників нутріціології — науки, що досліджує їжу та продукти, їх поживні елементи і їх взаємодію.

Під енергетичною цінністю продукту мається на увазі кількість енергії, яка вивільняється при травленні їжі в організмі людини. Але не лише отримання достатньої кількості енергії важливо для підтримки життєдіяльності організму, а і дотримання певного балансу білків, вуглеводів та жирів у щоденному харчуванні.

Кожен макронутрієнт відіграє у цьому свою функцію. Основним джерелом амінокислот виступають білки, необхідні для оновлення та обслуговування клітин. Жири використовуються для засвоєння основних вітамінів, побудові клітин шкіри та судин, та є будівельним матеріалом більшості гормонів. Обмін жирів та білків регулюється вуглеводами [3]. Вони ж є основним джерелом енергії.

Таким чином некоректне забезпечення основними харчовими елементами організму, їхня недостатня кількість чи перевантаження, суттєво впливає на здоров'я та функціональність людини.

Основних нутрієнти поділяються на макро- та мікронутрієнти та мають власні підгрупи (рис. 1.1).

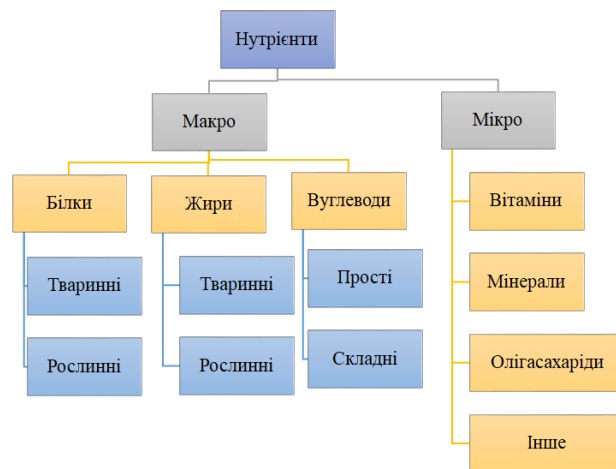


Рис. 1.1. Класифікація основних нутрієнтів, що входять до складу продуктів

Розвиток нутриціології в Україні знаходиться у перехідному стані від заміни усталених за радянськими часами практик, зокрема дієтичних столів за Певзнером, до загальносвітових напрямків [4].

Сучасна концепція здорової дієти полягає у збалансованому вмістом макронутрієнтів (білків, жирів, вуглеводів, харчових волокон), мікронутрієнтів, тобто вітамінів, і за загальною калорійністю харчуванні. При індивідуальній непереносимість певних продуктів, ефективним методом лікування вважається елімінаційна дієта. Так, при несприйнятті організмом лактози, необхідне дотримання безлактозної дієти з відмовою від молочних продуктів тваринного походження, а також низькі продуктів, що їх містять. Як альтернативно джерело кальцію при цьому можуть виступити зелені овочі, капуста, шпинат броколі. Таким чином як варіант лікування може розглядатися додаткове призначення альтернативних джерел нутрієнтів.

Для деяких хвороб дотримання пов'язаної дієти є критично необхідним для підтримки організму та запобігання погіршення симптомів. При цукровому діабеті другого типу важливо суттєве обмеження продуктів, що містять швидкі вуглеводи, таких як цукор, субпродукти, макаронні вироби та інше. При підвищеному артеріальному тиску рекомендовано більше урізноманітнення

щоденної дієти фруктами, овочами, продуктами, що містять харчові волокна. Для білка в організмі вживати нежирне м'ясо риби та птиці. При цьому треба обмежити споживання солі, кофеїну, технологічно обробленої їжі [5].

Певні загальні рекомендації лікувальних дієт можуть використовуватися і для профілактики й побудові плану харчування здорової людини. Зокрема зменшення споживання швидких вуглеводів та трансжирів є загальносвітовою рекомендацією для зменшення ризику виникнення багатьох хвороб.

Залежно від дієти співвідношення поживних речовин може бути змінено у ту чи іншу сторону. Але сильне відхилення від норми також є небезпечним і має відбуватися тільки за призначенням лікарів.

Вибір тої чи іншої дієти має прийматися спільно пацієнтом та лікарем з урахуванням персональних особливостей.

1.2. Норми поживних речовин та обмеження продуктів при дієтичному харчуванні

За рекомендацією Всесвітньої організації охорони здоров'я (далі ВООЗ) пропонується алгоритм оцінки енергетичних витрат, заснований на статі та віку людини, її ваги та рівні фізичної активності [6]. У табл. 1.1 та 1.2 представлені дані для розрахунку основного рівня обміну та фактору фізичної активності.

Таблиця 1.1

Розрахунок рівня основного обміну

Чоловіки	18 - 30 років	$240 \text{ ккал/день} \cdot (0.0630 \times \text{маса тіла, кг} + 2.8957)$
	31 - 60 років	$240 \text{ ккал/день} \cdot (0.0484 \times \text{маса тіла, кг} + 3.6534)$
Жінки	18 - 30 років	$240 \text{ ккал/день} \cdot (0.0621 \times \text{маса тіла, кг} + 2.0357)$
	31 - 60 років	$240 \text{ ккал/день} \cdot (0.0342 \times \text{маса тіла, кг} + 3.53707)$

Фактор фізичної активності

Малорухливий спосіб життя, низька фізична активність	1,3
Помірна фізична активність, регулярні навантаження - 30 хв. - 60 хв. на день	1,5
Висока (інтенсивні регулярні навантаження або фізична робота > 60 хв. на день)	1,7

Визначення загальних енергетичних витрат відбувається за наступною формулою: Рівень Основного Обміну (див. табл. 1.1) · Фактор Фізичної Активності (див. табл. 1.2).

Таким чином, чоловік у 30 років із масою тіла 70 кг має основний обмін близько 1753 ккал на добу. При середньому рівні фізичного навантаження його загальні енергетичні витрати дорівнюють $1753 \text{ ккал} \cdot 1.5 = 2629.5 \text{ ккал}$. Саме цю кількість енергії чоловіку необхідно отримувати для підтримки здорової і стабільної маси тіла.

Варто відмітити, що попередні розрахунки підходять для людини з вагою тіла, що знаходиться у межах норми.

При зниженій та надмірній вазі рекомендована кількість калорій може відрізнятись, залежно від рівня відхилення. У середньому чоловіки потребують близько 2200-2700 ккал./добу, а жінки – 1800-2200 ккал./добу, а діти близько 1500-2000 ккал./ добу.

Для визначення який тип ваги має людина одним з найбільш простих та об'єктивних способів є розрахунок значення індексу маси тіла (далі — ІМТ) [7].

Він розраховується за формулою (1.1):

$$\text{ІМТ} = \frac{m}{h^2}, \quad (1.1)$$

де

m – маса тіла у кілограмах;

h – зріст людини у метрах.

Якщо значення ІМТ знаходяться у межах 18,5-24,9 кг/м² це вважається нормальною масою тіла. Значення менше є зниженим ІМТ та сигналізує про нестаток харчування. Значення від 25 до 29,9 кг/м² вказують на надмірну вагу, а більше за 30 кг/м² на ожиріння I ступеня та необхідність перегляду харчових навичок та зниження рівня калорій, що потребляються

Для формування здорової дієти важливий не тільки розподіл білків, жирів та вуглеводів, але і їх підтипів.

Білки поділяються на рослинні, що містяться у бобових чи горіхах, та тваринні — з м'яса, сирів, риби, яєць. Вони виконують структурну функцію у побудові клітин організму. Для отримання найоптимальнішого співвідношення амінокислот, особливо незамінних, рекомендоване якнайбільше розширення різноманіття білкових продуктів у харчуванні, як рослинних, так і тваринних.

Під час окислення вуглеводів клітини отримують енергію, тож основна їх функція — енергетична. Для одного грама вуглеводу цей процес призводить до виділення в середньому 4 кілокалорії.

Основна частка жирів міститься у оліях: оливкових, кунжутних, гарбузових, та продуктах тваринного походження: у м'ясі та кисломолочних продуктах. Оптимальне співвідношення між цими типами є близько 70% жирів тваринного походження до 30% рослинного. При використанні свіжого рослинного масла без термічної обробки до організму поступають біологічно цінні речовини, особливо поліненасичені жирні кислоти.

Окремої уваги заслуговує споживання трансжирів, різновиду ненасичених жирів, серед яких небезпечні ті, що були отримані шляхом гідрогенізації під час промислового виробництва з рідких рослинних олій. Їх часте вживання може збільшити ризики низки захворювань, зокрема серцево-судинних, сприяє утворенню бляшок холестерину.

Вуглеводи поділяються на складні, до яких входить крохмаль у складі круп, хліба, картоплі, та прості: глюкоза, цукор. При споживанні вуглеводів оптимально отримувати їх у співвідношенні 4:1, з наданням переваги складним.

Важливо також різноманіття продуктів: від сортів хліба, різних круп, свіжих овочах та фруктах.

Згідно рекомендацій ВООЗ нормальний розподіл макронутрієнтів для дорослої людини складає:

- від 20 до 35% енергії з жиру;
- від 10 до 35% енергії з білку;
- від 45 до 65% енергії з вуглеводів [8].

Окислення грама білка або вуглеводу призводить до виділення близько 4 кілокалорій енергії. Окислення грама жирів до 9 кілокалорій відповідно.

На модель харчування також впливає наявність певних хвороб та вибір конкретного типу дієти.

У ході дослідження було розглянуто низьку основних лікувально-оздоровчих або підтримуючих дієт та їх характеристики.

Низьковуглеводна дієта передбачає обмеження споживання вуглеводів до 10%-25% від добової норми калорій. При цьому близько 40%-50% від загальної калорійності має приходитися на білки, 30%-40% на жири.

Рекомендується надання переваги продуктам з середнім та низьким глікемічним індексом, який є показником швидкості підвищення рівню цукру у крові. Таким чином дієта пропонує відмову від вживання простих вуглеводів, що містяться у цукру, рафінованих продуктах харчування, білому рисі та кашах швидкого приготування [9].

Завдяки обмеженню споживання вуглеводів ця дієта може знизити ризики цукрового діабету другого типу. Але надмірні обмеження можуть призвести до негативних наслідків та потребує консультації лікаря та дієтолога.

Високобілкова дієта — фокусується на збільшенні споживання білків у раціоні та зменшення споживання швидких вуглеводів та трансжирів. Цей підхід часто застосовується для підтримання м'язової маси, схудненню та загальному поліпшенню фізичного здоров'я. Рекомендується обирати здорові джерела білків, такі як м'ясо з низьким вмістом жирів, риба, яйця, молочні продукти, бобові, горіхи та насіння. Початковий раціон цієї дієти

зосереджується на отриманні близько 30% калорій від білка, 30% з жирів та 40% з вуглеводів [10].

Дієта для контролю артеріальної гіпертензії виникла у результаті багатьох досліджень та її дотримання вважається корисним навіть для здорових людей. Основним напрямком дієти є обмеження вживання солі до 4-5 грам на добу. Обмеження стосується не тільки додавання солі у домашні страви, а і зменшення споживання технологічно оброблених харчових продуктів. Не рекомендовано вживання тваринних жирів і продуктів, що містять швидкі вуглеводи. Передбачено високий рівень споживання овочів та фруктів, нежирних кисломолочних продуктів та морепродуктів. Баланс білків, вуглеводів та жирів знаходиться у межах 18%, 55% та 27% від добової калорійності відповідно [11].

Дієта при цукровому діабеті другого типу. Так як причиною виникнення цієї хвороби є неспроможність організму належним чином обробляти інсулін, що виробляється в організмі, ключову роль у її лікуванні відіграє здорове харчування та належний рівень фізичної активності. Рекомендованим розподілом щоденного раціону має бути підтримка наступного балансу: вуглеводи 55-60%, жири 20-25%, білки 15-20% та виключення з дієти простих рафінованих вуглеводів [12-14].

Середземноморська дієта характеризується набором продуктів, що містять багато харчових волокон, антиоксидантних сполук і протизапальних ефектів, що призводить до зменшення ризику виникнення низки хвороб, серед яких серцево-судинні захворювання, хвороби Альцгеймера, цукровий діабет другого типу [15].

Список обмежень дієти включає червоно м'ясо, жирні молочні продукти, цукор. Надається перевага фруктам та овочам, які складають половину щоденного раціону, злакам, бобовим, горіхам та насінню у раціоні, корисних жирам. Важливим джерелом останніх виступає оливкова олія. Норми поживних речовин у цієї дієти зазвичай складає 50% вуглеводів, біля 15% білків та 35% жирів [16].

Вегетаріанська дієта має багато варіацій: від певного обмеження у споживанні м'ясних страв до повної відмови від будь-яких продуктів тваринного походження. Лактоововегетаріанська дієта дозволяє споживання молочних продуктів, яєць та виключає лише м'ясо.

Дотримання вегетаріанської дієти, особливо у комбінації з здоровою фізичною активністю, може призвести до зменшення вірогідності хвороб серця, ожиріння, цукрового діабету другого типу. При цьому жорстке слідкування потребує додаткових ресурсів для отримання вітаміну B12, а відсутність у раціоні молочних продуктів може призвести до дефіциту вітаміну D й кальцію в організмі.

Важливою частиною будь-якої дієти є індивідуалізація та консультація з фахівцем з харчування чи лікарем, особливо перед тим, як змінювати свій звичайний раціон. Кожна особа має унікальні харчові потреби, і підходячий режим харчування повинен враховувати їхні індивідуальні особливості, стан здоров'я та фізичну активність.

1.3. Проблема оптимальної закупівлі продуктів відповідно до потрібного раціону харчування

Полиці сучасних магазинів пропонують багатий вибір товарів, що одночасно спрощує і ускладнює підбір товарів для формування здорового раціону. Майже завжди можна знайти потрібні продукти, обрати найкращі або найбільш відповідні за характеристиками. У той же час, різноманіття має і негативний бік. Вибір не завжди здається очевидним. Техніки агресивного маркетингу спокушають потребляти більше і без наявності плану дуже просто вийти за рамки або не розрахувати можливості та витратити більше, ніж потрібно.

Бюджет на харчування заслуговує окремої уваги. Війна, нестабільність економічної ситуації призводять до різких змін вартості продуктів [17]. За останні п'ять років, вартість борошна, одного з основних продуктів, зросла більше ніж у два рази (рис.1.2).

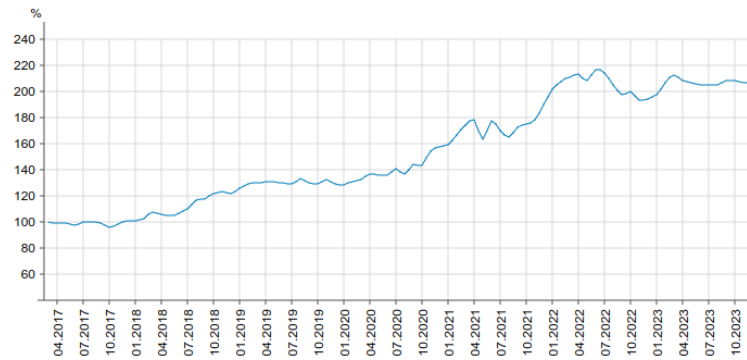


Рис. 1.2. Динаміка зміни індексу середньомісячних цін на борошно згідно сайту міністерства фінансів

На осінь 2023 року вартість базового продуктового кошику дорослої людини може складати біля 50% мінімальної заробітної плати [18-19]. Через це необачна закупівля продуктів може призвести до недоотримання потрібної кількості білків, жирів, вуглеводів та вітамінів. Покупка великої кількості продуктів, що містять певні типи поживних речовин, може призвести до того, що бюджету не вистачить на інший, необхідний для формування здорового харчування.

Також дотримання певного типу дієти вимагає додаткової уваги для підбору відповідного набору продуктів та формуванні здорового меню, що задовольняє обмеженням та надає потрібний баланс нутрієнтів.

Це все призводить до того, що планування закупівель, особливо при обмеженому бюджету, є комплексним процесом.

Таким чином при організації здорового харчування можна виділити наступні завдання та проблеми:

- закупівля продуктів згідно рекомендованої дієти;
- підтримка балансу нутрієнтів;
- планування бюджету на продукти у часи стрімких змін ринкових цін;
- ефективність закупівель при обмеженому бюджету.

1.4. Застосування автоматизації для плануванні продуктового кошика

Цифровізація та автоматизація торкнулася всіх сфер життя: від високоточних розрахунків космічних систем до вирішення побутових проблем, у тому числі плануванні.

Програмні додатки пропонують плани тренувань, навчання, дієт, дозволяють швидко сформувавши рішення при наявності багатьох умов та факторів. Вони не замінюють професійного тренера чи дієтолога, але допоможуть заощадити час та зіграти роль корисного помічника.

Автоматизації планування закупівель має взяти до уваги різні види дієт для формування списку продуктів та реалії ринка або ринків, для якого вона розрахована. Набір та вартість продуктового кошику різних країн вибір та сезонність товарів. Таким чином програмний додаток, створений для користувачів з Америки, не завжди може підійти для користувачів з України, через локальні особливості.

Завдяки програмним обчисленням можна автоматизувати процес індивідуального підрахунку норми калорій за характеристиками користувача. За вказаним обмеженням бюджету, типом дієти та її обмеженням додаток може обробити список продуктів та обрати з них ті, чия вартість, склад та співвідношення білків, жирів та вуглеводів зможуть оптимально задовольнити потреби користувача та обраний їм план харчування.

Користувач має можливість швидко виключити або додати групи продуктів за власними потребами. Наприклад, при несприйнятті лактози, прибрати з розрахунку молочні продукти, або при дотриманні вегетаріанської дієти — м'ясні вироби.

Перевагою використання автоматизації для планування продуктового кошика є:

- економія часу за рахунок перенесення обчислень та розрахунків до програмної системи;

- зручний вибір та перегляд особливостей різних дієт;
- отримання списку продуктів, що відповідає потрібної дієті;
- автоматизований контроль за співвідношенням та вмістом нутрієнтів у продуктів, що входять у список для закупівлі;
- отримання рекомендацій щодо збалансованому списку закупівель при обмеженні бюджету;
- отримання актуальних цін на продукти та їх харчовий склад;
- збереження результатів у електронному вигляді, завдяки чому знижується імовірність забути список закупівель вдома.

Недоліками автоматизованої системи є:

- обмеження кількості продуктів. Система не зможе врахувати все різноманіття продуктів, що представлені на полицях магазинах, і буде містити лише загальний їх набір;
- складність урахування всіх факторів приготування. Система бере дані для продуктів лише у певному стані, але в залежності від типу приготування харчовий склад продукту може змінитися за рахунок чого обчислення будуть неточні;
- автоматизована система не є заміною повноцінного дієтолога, а її рекомендації, особливо для лікувальних хвороб, мають бути узгоджені з лікарем.

1.5. Аналіз пов'язаних рішень

При дослідженні було розглянути та досліджено різні програмні додатки, що включають можливість генерації списків продуктів харчування та перегляду їх вартість.

Було виділено мобільний додаток для планування меню та списку покупок Lifesum: Healthy Eating & Diet. Він сфокусований на підрахунку калорій та поживних речовин, плануванню блюд на кожен день тижня та створенню відповідних списків покупок. Особлива увага приділяється дієтам

для зниження ваги та формуванні звичок здорового харчування. Додаток включає трекер мікронутрієнтів, пропонує вибір дієт серед яких: низьковуглеводна дієта, скандинавська, середземноморська дієта та низька інших. Він пропонує формувати плани харчування зі списком продуктів на тиждень згідно обраного типу дієти [20].



Рис. 1.3. Інтерфейс додатку Lifesum

Серед переваг рішення можна назвати багатий вибір різноманітних дієт та блюд, можливість слідкування за балансом нутрієнтів та калорій, планер списків продуктів та зручний, яскравий інтерфейс (рис.1.3).

На жаль, додаток не має локалізації українською та при плануванні списку продуктів орієнтується на американський та європейський ринок. Таким чином, хоча Lifesum є зручним помічником для здорового харчування, планування списку продуктів не є достатньо зручним в Україні, коли метою є саме контроль та планування бюджету. Також у списку дієт додатка відсутні ті, що мають лікувальну направленість при цукровому діабеті чи гіпертонії.

Отримати інформацію щодо вартості продуктів основного продуктового кошику можна на сайті міністерства фінансів України.

Там представлено багато корисної інформації, у тому числі індекси цін на продукти харчування. Сайт пропонує дані про середню вартість певного найменування продукту. Тут можна подивитися статистику та підрахувати скільки буде коштувати продуктивий кошик покупця (рис. 1.4).

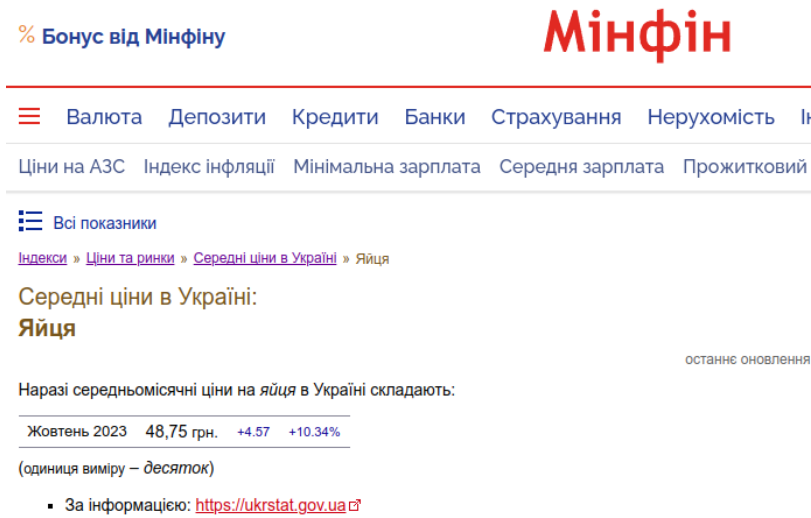


Рис. 1.4. Перегляд інформації про середньомісячну вартість яєць

Даний сайт представляє лише статистичні дані. Таким чином, його дані можуть бути використані для отримання актуальної інформації про вартість продуктів, вона потребує додаткової обробки, прив'язки продуктів до їх енергетичної цінності та вже подальшому використанні у програмному додатку автоматизації планування [17].

1.6. Дослідження вимог до програмної системи

Програмний додаток для автоматизації планування дієти має за введеними даними користувача виконувати моделювання та оптимізацію набору продуктового кошику з урахуванням балансу нутрієнтів та індивідуальних обмежень користувача.

Таким чином його функціонал має задовольняти наступні вимоги:

1. Заповнення вхідних даних для обчислення: обмеження бюджету, період для якого треба виконати обчислення.
2. Введення даних користувача: статі, віку, ваги, зросту, рівню фізичної активності.
3. Розрахунок добової рекомендованої норми калорій за введеними даними.

4. Вибір бажаної дієти та додаткових обмежень щодо споживання продуктів.

5. Кастомізація списку продуктів, на основі яких будуть виконуватися розрахунки.

6. Виконання моделювання продуктового кошику, що відповідає обраному типу дієти та параметрах.

7. Вивід результатів моделювання у вигляді списку продуктів та статистичної інформації щодо співвідношення макронутрієнтів та енергетичної цінності продуктового кошику.

8. Експорт та завантаження результатів моделювання.

9. Адміністрування програмного додатку: управління списками продуктів, обмежень та дієт, управління списками користувачів.

Програмна система має мати три ролі користувачів:

– кінцевого користувача, що задає дані для розрахунків та отримує результати;

– менеджера, що контролює актуальність списків продуктів та обмежень;

– адміністратора, що контролює користувачів програмної системи.

Також користувачі системи розподіляються на авторизованих та неавторизованих користувачів, перші з яких мають мати можливість збереження введених даних та перегляду розрахованих раніше списків продуктів.

Програмний додаток має бути представлений у вигляді клієнт-серверного веб-додатку. Він має підтримувати адаптацію до різних типів пристроїв та операційних систем. Веб-додаток має працювати у всіх сучасних браузерах та мати зрозумілий, зручний інтерфейс. Виконання обчислень має виконуватися на боці серверу з введенням та виведенням результатів на боці клієнта.

1.7. Висновки

Сучасна дієтологія спирається на персоніфікованому підході у формуванні здорового харчування, враховуючи мікробіологічні, генетичні та багато інших факторів. Важливим є повноцінне та різноманітне харчування з отриманням достатньої кількості всіх типів мікро- та макронутрієнтів.

Залежно від дієти може змінюватися список обмежень та співвідношення білків, жирів, вуглеводів.

Так як підготовка списку закупівель відповідний потрібній дієті і плану бюджету може виявитися комплексним завданням, автоматизація цього процесу дозволить зекономити час, сформувати корисні харчові звички та полегшить контроль за добовим вживанням калорій та нутрієнтів.

Аналіз показав, що існує велика кількість додатків для генерації дієт, але їм не вистачає прив'язки до реалій українського ринку, через що результати їх роботи можуть виявитися не релевантними щодо вартості та наборів продуктів.

Таким чином розробка автоматизованої системи планування продуктового кошику, що буде пропонувати інформацію про харчову цінність та ціну продуктів, а також надавати можливість обирати різні типи лікувальних та профілактичних дієт з кастомізацією вхідних даних та обмежень, є доцільною. Даний програмний додаток буде широко доступним завдяки роботі через браузер та адаптивність інтерфейсу.

Наприкінці розробки програмного додатку мають бути проведено тестування та порівняння результатів робіт додатку з загальними рекомендаціями щодо складів дієт та зроблені висновки його ефективності моделювання.

РОЗДІЛ 2

ДОСЛІДЖЕННЯ ТА ПОБУДОВА МАТЕМАТИЧНОЇ МОДЕЛІ

2.1. Підготовка вхідних даних

Перед формування математичної моделі генерації списку продуктів для закупівель потрібно провести обчислення та підготовку вхідних даних.

Підготовка даних включає 3 етапи:

1. Розрахунок добової норми калорій людини.
2. Визначення балансу макронутрієнтів відносно обраної дієти.
3. Підготовка списку продуктів та обмежень.

За введеними користувачем даними: статтю, віком, вагою, зростом та рівнем фізичної активності розрахуємо його добову норму калорій та ІМТ.

Для отримання індексу маси тіла використаємо формулу (1.1). Якщо ІМТ знаходиться у межах норми 18,5-24,9 кг/м², розрахуємо рівень основного обміну за формулою Міффіліна–Сан-Жеора [21]. Для чоловіків розрахунок буде мати наступний вигляд, представлений у формулі (2.1).

$$BMR = 10 \cdot W + 6.25 \cdot H - 5 \cdot Y + 5, \quad (2.1)$$

де

BMR – рівень основного метаболізму, ккал;

W – маса людини, кг;

H – зріст людини, см;

Y – вік людини, рік.

Для жінок формула Міффіліна–Сан-Жеора має різницю у значенні останньої константи, як показано у формулі (2.2).

$$BMR = 10 \cdot W + 6.25 \cdot H - 5 \cdot Y - 161. \quad (2.2)$$

Якщо ІМТ людини перевищує чи недовідає нормальних значень, добова норма калорій рекомендується згідно значень табл. 2.1.

Таблиця 2.1

Співвідношення денної потреби в енергії відносно ІМТ без урахування рівня фізичної активності

Тип	Значення ІМТ	Рекомендована кількість калорій, ккал / кг / добу
Знижений ІМТ	< 18,5 кг/м ²	25
Нормальний ІМТ	18,5-24,9 кг/м ²	20
Надмірна вага тіла	25-29,9 кг/м ²	17
Ожиріння I ступеня	30-34,9 кг/м ²	15
Ожиріння II ступеня	35-39,9 кг/м ²	15
Морбідне ожиріння	> 40 кг/м ²	15

Для отримання добової норми калорій потрібно помножити рівень основного метаболізму на індекс фізичної активності, дивись табл. 2.2.

Таблиця 2.2

Індекс активності в залежності від рівню фізичної активності людини

Рівень фізичної активності	Індекс фізичної активності
Відсутність фізичних навантажень	1,2
Легкі фізичні вправи, гімнастика від 1 до 3 разів на тиждень.	1,375
Середні фізичні навантаження 3-5 разів на тиждень.	1,55
Активні тренування від 6 до 7 разів на тиждень.	1,725
Тренування більше 1 разу на день з важкими силовими вправами. Важка робота, пов'язана з фізичною працею.	1,9

Для отримання добової норми калорій K потрібно помножити рівень основного обміну на Індекс фізичної активності.

Наступним кроком є визначення мінімального значення білків, жирів та вуглеводів для обраної дієти, дивись табл. 2.3.

Таблиця 2.3

Норми макронутрієнтів для різних типів харчування

Назва дієти	Кількість макронутрієнтів відносно добової норми калорій
Збалансована дієта	Вуглеводи: 45-65% Білки: 10-35% Жири: 20-35%
Низьковуглеводна дієта	Вуглеводи: 10-25% Білки: 40-50% Жири: 30-40%
Високобілкова дієта	Вуглеводи: 40%-45% Білки: 30-35% Жири: 25-30%
Дієта для контролю артеріальної гіпертензії	Вуглеводи: 50-55% Білки: 18-22% Жири: 23-27%
Дієта при цукровому діабеті другого типу	Вуглеводи: 55-60% Білки: 15-20% Жири: 20-25%,
Середземноморська дієта	Вуглеводи: 45-50% Білки: 15-20% Жири: 30-35%
Вегетаріанська дієта	Вуглеводи: 45-65% Білки: 10-35% Жири: 20-35%

Останнім етапом підготовки є обробка списків продуктів та підготовка їх для обчислень: визначення вартості, калорійності та вмісту макронутрієнтів на одиницю продукту. Список продуктів фільтрується за обмеженнями: виключаються продукти або їх групи, що заборонені у обраному типі дієти.

Нарешті для кожного продукту прив'язуються обмеження щодо його мінімальної та максимальної дозволеної кількості.

2.2. Дослідження математичної моделі

Змоделюємо ситуацію, коли людина вирішила дотримуватися певної дієти та має намір зібрати продуктовий кошик на певний період часу t . При цьому існує обмеження на максимальну вартість кошику, що дорівнює s_{max} .

Кількість продуктів у кошику позначимо через x_i , $i = 0, 1, \dots, n - 1$, кожен продукт має власну ціну за одиницю p_i . Загальну вартість всіх продуктів позначимо через s .

Метою моделювання є мінімізація цільової функції (2.3), що визначає суму всіх продуктів продуктового кошику.

$$\min_x \sum_{i=0}^{n-1} p_i x_i = s, s > 0. \quad (2.3)$$

Для повноцінного функціонування організму ця людина має отримувати з їжі за добу певну кількість енергії E , що була розрахована у попередньому розділі. Таким чином додаються наступні обмеження у формулі (2.4).

$$\begin{cases} \sum_{i=0}^{n-1} e_i x_i = W \\ s \leq s_{max} \end{cases}, \quad (2.4)$$

де

e_i – кількість енергії на одиницю продукту, ккал;

$W = t \cdot E = const$ – потреби в енергії за певний термін;

E – затрати енергії на добу;

t – період часу, діб;

s – загальна вартість продуктів;

s_{max} – максимальна допустима вартість продуктів.

Варто передбачити ситуацію, коли програма може обрати лише один або кілька найбільш дешевих та калорійних продуктів у якості результату. Таким чином додається обмеження на мінімальну та максимальну кількість вживання продукту у певний період. Обмеження представлено у формулі (2.5)

$$x_{i \min} \leq x_i \leq x_{i \max}, \quad (2.5)$$

де

x_i – кількість продукту;

$x_{i \max}$ – мінімальна кількість продукту за певний період;

$x_{i \min}$ – мінімальна кількість продукту за певний період.

Кожна дієта має власне рекомендоване співвідношення макронутрієнтів, необхідний для забезпечення повноцінної роботи організму. Таким чином останніми обмеженнями моделі є обмеження на мінімальну та максимальну кількість білків, жирів та вуглеводів, що мають бути отримані у результаті розрахунків. Представимо ці обмеження у формулі (2.6).

$$\left\{ \begin{array}{l} pr_{min} \leq \sum_{i=0}^{n-1} x_i pr_i \leq pr_{max} \\ c_{min} \leq \sum_{i=0}^{n-1} x_i c_i \leq c_{max} \\ f_{min} \leq \sum_{i=0}^{n-1} x_i f_i \leq f_{max}, \end{array} \right. \quad (2.6)$$

де

pr_i – кількість білків на одиницю продукту;

pr_{max} – максимальна кількість білків за заданий період;

pr_{min} – мінімальна кількість білків за заданий період;

c_i – кількість вуглеводів на одиницю продукту;

c_{max} – максимальна кількість вуглеводів за заданий період;

c_{min} – мінімальна кількість вуглеводів за заданий період;

f_i – кількість жирів на одиницю продукту;

f_{max} – максимальна кількість жирів за заданий період;

f_{min} – мінімальна кількість жирів за заданий період.

Вирішення завдання полягає у знаходженні списку продуктів з мінімізацією їх загальної вартості $s \rightarrow \min$ та отриманні максимальної потрібної кількості енергії з урахуванням обмежень, що задаються у формулах (2.4), (2.5) та (2.6).

2.3. Вибір інструмента оптимізації набору продуктів

Для програмної реалізації визначеної математичної моделі було досліджено бібліотеки, що пропонують функціонал оптимізації скалярної функції. Серед вимог до інструмента оптимізації була надійність, можливість роботи з великим набором даних, зручність інтеграції. Було розглянуто низьку бібліотек, що надають подібний функціонал та виділено наступні.

Бібліотека Python LMfit включає інструменти для мінімізації нелінійного методу найменших квадратів і підгонки кривої. Сфокусована на гнучких алгоритмах оптимізації та зручності використання. Ключовою особливістю бібліотеки є особливістю робота зі змінними у процедурах мінімізації. Серед недоліків LMfit можна відмітити її новизну та відсутність великої підтримки спільноти [22].

Однією з передових мов у сфері аналізу даних виступає мова програмування R, що пропонує середовище для аналізу, машинного навчання і статистичних обчислень. Вона містить бібліотеки для оптимізації, зокрема модуль `optimizer` для оптимізації цільової функції з обмеженнями. Ця мова має потужний математичний апарат, але з її недоліків можна відмітити чітку направленість, що не дозволяє використовувати R для більш широкого сектору

задач. Через це підключення оптимізаційних розрахунків має відбиватися через окремий модуль.

Бібліотека Python SciPy розроблена для роботи з науковими алгоритмами та обчисленнями, що включає методи оптимізації, інтерполяції, лінійної алгебри, обробку зображень. Серед переваг бібліотеки велика підтримка спільноти, відкрита ліцензія, великий набір функцій, що були розроблені та протестовані багатьма розробниками та науковцями [23]. Серед недоліків можлива повільність певних методів оптимізації, що потребують уважного формування умов та додаткових налаштувань.

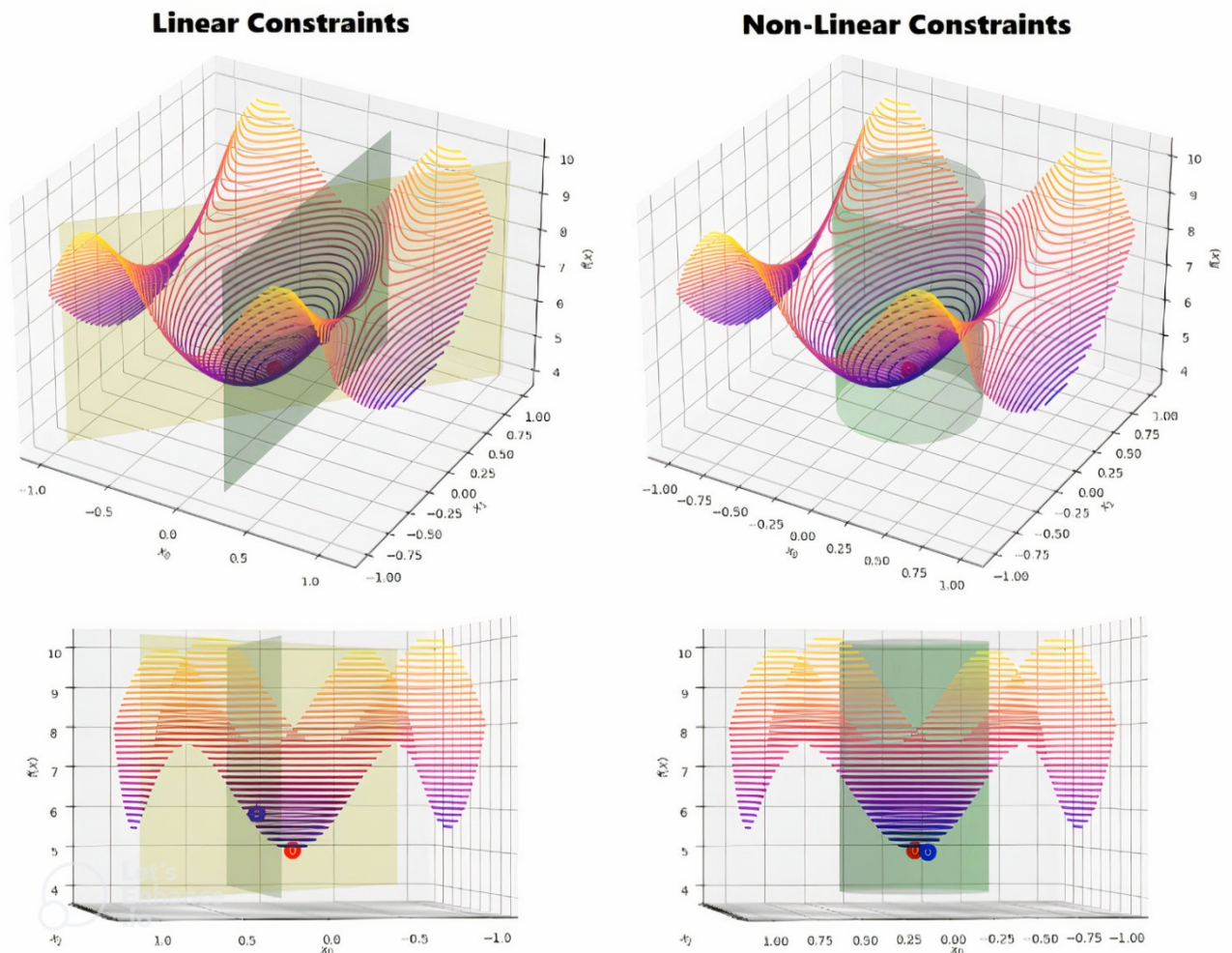


Рис. 2.1. Візуалізація методів оптимізації бібліотеки SciPy

У результаті досліджень було вирішено використовувати бібліотеку SciPy (рис.2.1), як зручну за співвідношенням надійності, швидкості та поєднанню з іншими модулями програмного забезпечення.

2.4. Мінімізація скалярної функції кількох аргументів при оптимізації набору продуктів

Метод послідовного квадратичного програмування SLSQP бібліотеки Python SciPy працює за алгоритмом: знайти $\min_x f(x)$ при обмеженнях (2.7):

$$\begin{aligned} c_j(x) &= 0, j \in \varepsilon \\ c_k(x) &\geq 0, k \in \tau \\ lb_i &\leq x_i \leq ub_i, i = 1, \dots, N, \end{aligned} \quad (2.7)$$

де

ε та τ – набори індексів, що містять обмеження рівності та нерівності;

c_j та c_k – функції обмеження;

lb_i – індивідуальне обмеження нижнього кордону значення x_i ;

ub_i – індивідуальне обмеження верхнього кордону значення x_i .

При виклику методу задається також цільова функція, допустима похибка обчислень, максимальна кількість ітерацій [24].

Метод SLSQP виконує послідовне вирішення задачі квадратичного програмування, які апроксимують задану функцію оптимізації. Для пошуку наближеного розв'язку надлишково-визначеної системи SLSQP використовує лінійний метод найменших квадратів.

Метод було обрано через наступні фактори:

- врахування особливостей реалізації типів даних у програмній системі, що підвищує ефективність та швидкість обчислень;
- функціонал для вирішення задачі оптимізації з можливістю задавати лінійні та нелінійні обмеження.

2.5. Висновки

У результаті аналізу функціональних та технічних вимогу було сформовано алгоритм підготовки та обробки вхідних даних перед виконанням оптимізації. Була сформована математична модель, мета якої — підвищення

ефективності закупівель продуктів харчування за рахунок зменшення витрат та отримання найбільш оптимального набору продуктів за енергетичними характеристиками та балансом білків, жирів та вуглеводів.

Було досліджено програмні методи реалізації процесу оптимізації та обрано бібліотеку Python SciPy, що пропонує гнучкий та ефективний математичний апарат для вирішення задач подібного напрямку. У якості метода оптимізації було обрано метод послідовного квадратичного програмування, який дозволяє формувати граничні та загальні обмеження для мінімізації цільової функції.

РОЗДІЛ 3

СТВОРЕННЯ ТА АНАЛІЗ РОБОТИ ВЕБ-ДОДАТКА ОПТИМІЗАЦІЇ НАБОРУ ПРОДУКТІВ

3.1. Архітектура веб-додатку оптимізації набору продуктів для збалансованого харчування

При розробці архітектури веб-додатку були проаналізовані функціональні вимоги й алгоритм виконання операцій для розрахунку продуктового кошика, та було прийняті наступні рішення:

- візуальний інтерфейс та сервер веб-додатку мають бути розділені на окремі сервіси для підвищення гнучкості та масштабованості системи;
- зв'язок між сервісами організувати буде відбуватися REST API;
- розділення інтерфейсів користувача та адміністратора за окремими адресами;
- ідентифікація авторизованого користувача системи відбувається за допомогою обміном JSON web-токен до заголовків запитів.

Було розглянуто основні ролі програмного продукту та виділено напрямки їх взаємодії з програмною системою:

- адміністратор програмної системи виконує контроль за користувачами, перегляду їх списків та видаленням користувачів;
- менеджер виконує контроль за списків продуктів та їх обмежень, а також запускає операцію автоматичного оновлення вартості продуктів;
- неавторизований користувач системи має можливість переглядати списки продуктів та обмежень за замовчуванням та виконувати операцію оптимізації продуктового кошика за введеними даними та перегляди її результати. Також він має можливість створювати профіль у системі;
- авторизований користувач має можливість зберігати індивідуальні налаштування списків продуктів, обмежень та переглядати раніше розраховані продуктові кошики.

На рис. 3.1 представлена UML-діаграма варіантів використання веб-додатка оптимізації продуктового кошика.

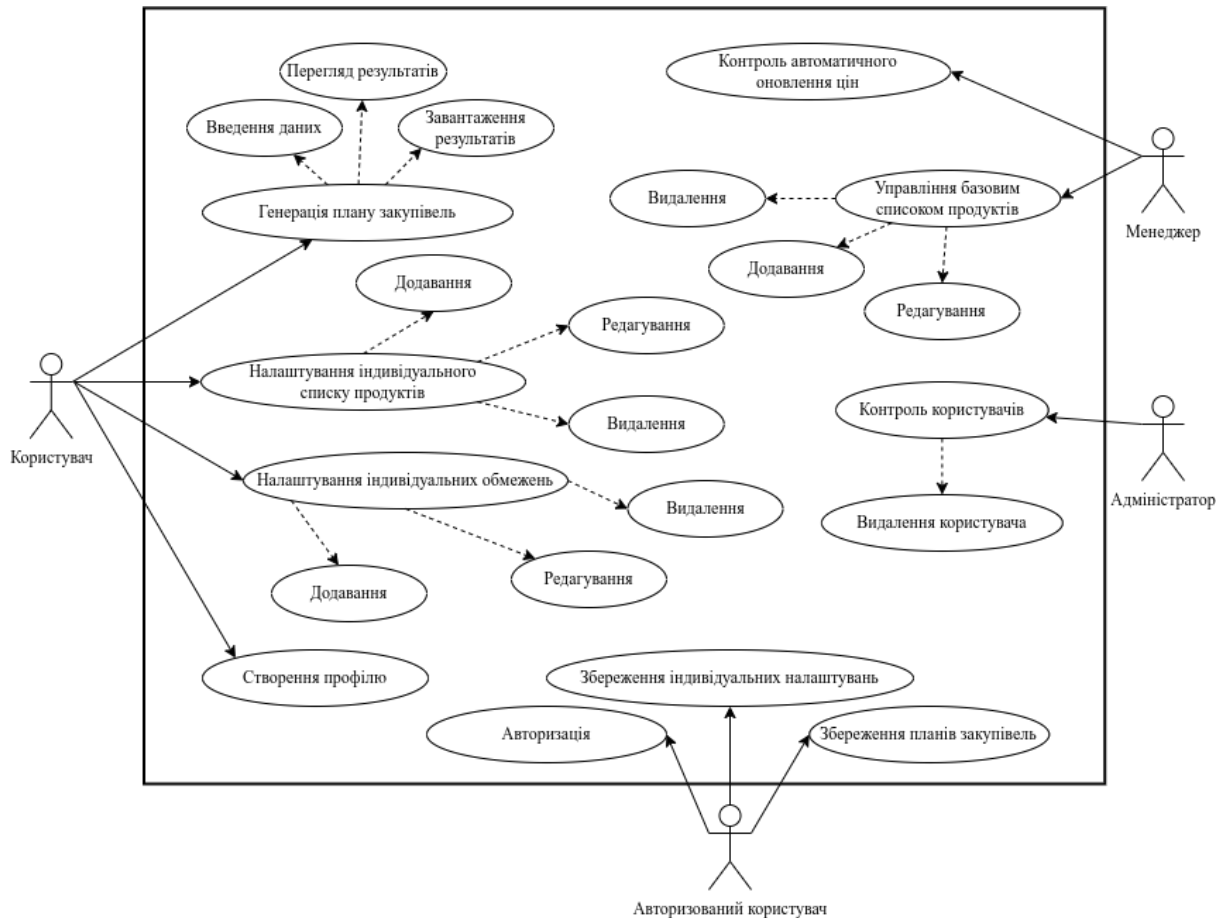


Рис. 3.1. UML-діаграма варіантів використання

Було виділено головні сутності системи та їх взаємозв'язки:

- Diet – дієта. Включає назву, опис та рекомендоване співвідношення нутрієнтів для даної дієти;
- Category – група, до якої відноситься продукт. Наприклад, «м'ясні продукти» або «Овочі»;
- Product – продукт харчування. Поділяється на три окремі сутності: узагальнений продукт, наприклад «картопля рожева»; продукт у магазині, що описує конкретний продукт в магазині, що має певну вартість та вагу; стан продукту, що містить інформацію про енергетичну характеристику продукту у різних станах приготування. Наприклад, «картопля рожева варена»;

– **Restriction** – обмеження споживання. Містить інформацію про мінімальну або максимальну рекомендовану кількість продукту для вживання на добу;

– **User** – користувач програмної системи. Його дані розділяються на два окремих блока: дані для авторизації та дані для проведення розрахунків;

– **Product Basket** – оптимізований продуктовий кошик. Містить список продуктів, їх кількість, вартість та енергетичну цінність.

Діаграма основних класів програмної системи представлена на рис. 3.1.

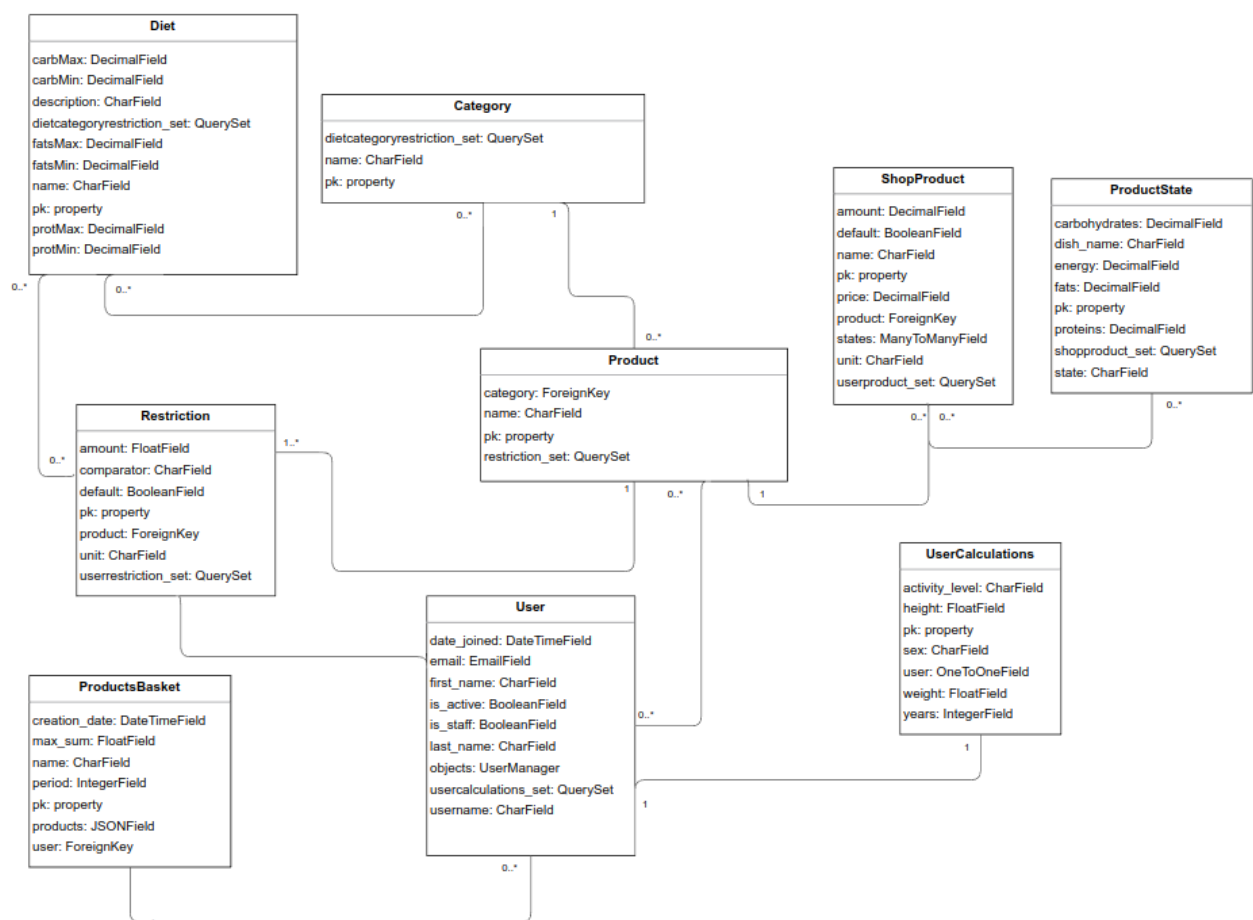


Рис. 3.2. Діаграма класів програмної системи

Так як виконання оптимізації відбувається за допомогою бібліотеки мови програмування Python, розробка сервісу буде базуватися на технології цієї ж мови для ефективного поєднання компонентів системи. Результати аналізу фреймворків дивись табл. 3.2.

Таблиця 3.1

Порівняння основних Python-фреймворків для розробки веб-додатків

Назва	Переваги	Недоліки
Django	<ul style="list-style-type: none"> – містить широкий набір інструментів для ефективно розробки веб-додатків; – включає вбудовані компоненти для роботи з базами даних, аутентифікації, адміністрування, шаблонами і багатьма іншими функціями; – включає Django Rest Framework для створення RESTful API. 	<ul style="list-style-type: none"> – через об'ємність, не завжди підходить для невеликих додатків; – вимагає більше часу для налаштування.
Flask	<ul style="list-style-type: none"> – легкий та мінімалістичний фреймворк, що містить базовий функціонал для розробки веб-додатків. – надає більшу гнучкість у розробці; – має менший розмір та споживає менше ресурсів, що робить його дуже зручним для невеликих проєктів. 	<ul style="list-style-type: none"> – не має вбудованих компонентів для багатьох завдань, може виникнути необхідності використовувати сторонні бібліотеки; – необхідність самостійної організації архітектури додатку.
FastAPI	<ul style="list-style-type: none"> – надає високу продуктивність та швидкість завдяки використанню типізації Python та автоматичному створенню документації API; – вбудована підтримка асинхронного програмування, що робить його ідеальним для реалізації асинхронних API. 	<ul style="list-style-type: none"> – новий фреймворк і доки не має такої великої кількості сторонніх бібліотек і розширень, як Django або Flask; – через наявність багатьох функцій може бути не дуже зручним для маленьких проєктів.

Після аналізу було вирішено задіяти Django веб-фреймворк через зручну архітектуру, налаштування якої дозволяє ефективно розділити шари представлення, обробки та збереження даних, а також потужну вбудовану систему роботи з сутностями бази даних, авторизацією та адмініструванням.

Для взаємодії з базою даних Django має вбудований ORM, що дозволяє представляти таблиці у вигляді програмних класів та виконувати операції на

ними через методи, а не прямі SQL-запити, що підвищує ефективність та безпечність роботи.

Фреймворк Django реалізує шаблон Model – Template – View, далі MTV.

Частина Model описує шар доступу до даних: перевірка вхідних значень, обробка та взаємодія з базою даних.

Частина Template описує шар відображення даних на сторінці, або передачі до клієнта.

Частина View відповідає за бізнес-логіку додатку. Тут контролюється доступи до моделей, виконуються обчислення і формуються у відповідний шаблон. Цей шар зв'язує між собою Model та Template.

Для організації ефективної взаємодії між API сервера та клієнтом було вирішено використати Django Rest Framework, що представляє інструменти для серіалізації та десеріалізації даних: для конвертації даних з об'єктів Python у JSON або інший формат для відправки через мережу.

Також Django Rest підтримує реалізацію методів автентифікації за допомогою JSON веб-токенів.

Разом з Django Rest з архітектури веб-додатку не буде виключений шар Template, так як відображення буде відбуватися на окремому клієнті, а сервер буде відповідати лише за обробку, відправку та збереження даних [25-26]. Таким чином взаємодія компонентів буде подібною до представленої на рис. 3.3.

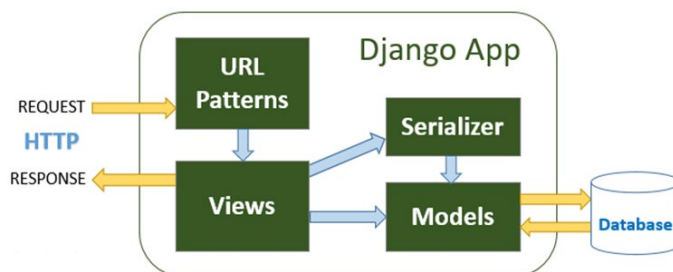


Рис. 3.3. Взаємодія компонентів серверу

Для збереження даних користувачів, продуктів та розрахунків було обрано СУБД PostgreSQL. Дана СУБД підтримує принципи ACID (Atomicity,

Consistency, Isolation, Durability) роботи з даними, що робить її стабільним та цілісним рішенням для систем, де важлива цілісність інформації.

Серед недоліків СУБД можна визначити високу вимогу до ресурсів та підвищення витрат на сервер, а також складність налаштуванні та управлінні порівняно з іншими, менш потужними СКБД. Остання особливість нівелюється використанням ORM, що спрощує та стандартизує методи роботи з СКБД.

Для побудови клієнт-частини додатку було обрано фреймворк JavaScript Angular, що пропонує зручну модульну архітектуру проекту, механізми маршрутизації та потужну екосистему роботи з реактивними формами.

Додаток Angular розділяє елементи інтерфейси на окремі компоненти, що поєднуються між собою та формують деревоподібну структуру, дивись рис. 3.4.

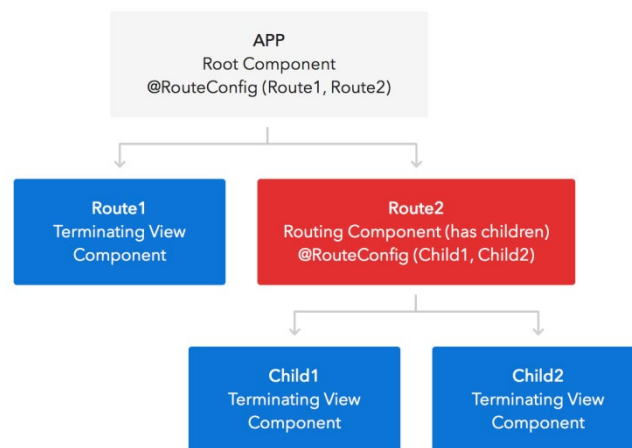


Рис. 3.4. Структура компонентів Angular проекту

Між елементами дерева, представленням та логікою компонента працює двосторонній зв'язок, що дозволяє передавати дані від батьків до дітей та навпаки, а також при зміні даних автоматично оновлювати вміст сторінок.

Для підвищення ефективності роботи та оновлення даних до проекту було підключено окрему бібліотеку управління станами, що реалізує Redux-архітектуру. У основі цього підходу є виділення окремого сховища даних, глобального у рамках проекту чи модуля. Оновлення та зміна даних у сховищі відбувається при реакції на певні події. У свою чергу компоненти підписуються

на зміну даних у сховищі та при їх отриманні виконуються оновлення інтерфейсу. Зручність такого підходу у відокремленні логіки роботи з запитами до сервера, збереження та представлення даних, рис. 3.5.

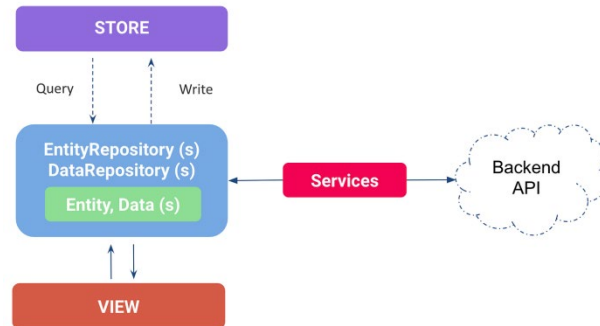


Рис. 3.5. Процес взаємодії елементів при Redux архітектурі

Для опису структури сторінки Angular використовує HTML, мову розмітки гіпертексту, яка є стандартом для побудови web-інтерфейсів. Сторінка складається з набору HTML-тегів, що поєднуються між собою у деревоподібну структуру та складають об'єктну модель документа DOM.

Для стилізації інтерфейсів web-сторінок проект використовує каскадні таблиці стилів CSS та препроцесор SCSS, що підвищує зручність опису ієрархії класів, додає можливості написання окремих функцій, змінних та умов.

Опис логічної частини клієнта виконується мовою TypeScript, яка є надбудовою над JavaScript, скриптової мови програмування, поширений для написання сценаріїв взаємодії з веб-сторінками. TypeScript додає можливості типізації та більш класичного використання класів за ООП принципами. Завдяки типізації підвищується зручність контролю та самодокументація коду, а також можливості рефакторингу.

Взаємодія клієнта та сервера відбувається за допомогою обміну REST-запитами, що реалізують методи GET, POST, PUT, DELETE для отримання, створення, редагування та видалення даних відповідно.

Для ідентифікації авторизованого користувача у заголовка запитів відправляється JWT токен, який клієнт отримує після проходження процесу авторизації. Токен містить інформацію про закодований об'єкт, алгоритм

хешування, корисне навантаження та підпису. Завдяки перевірці цих частин та контролю хешу, що зберігається на сервері та отримується з токена відбувається перевірка коректності JWT токена [27].

Повна схема взаємодії клієнта та сервера представлена на рис. 3.6.

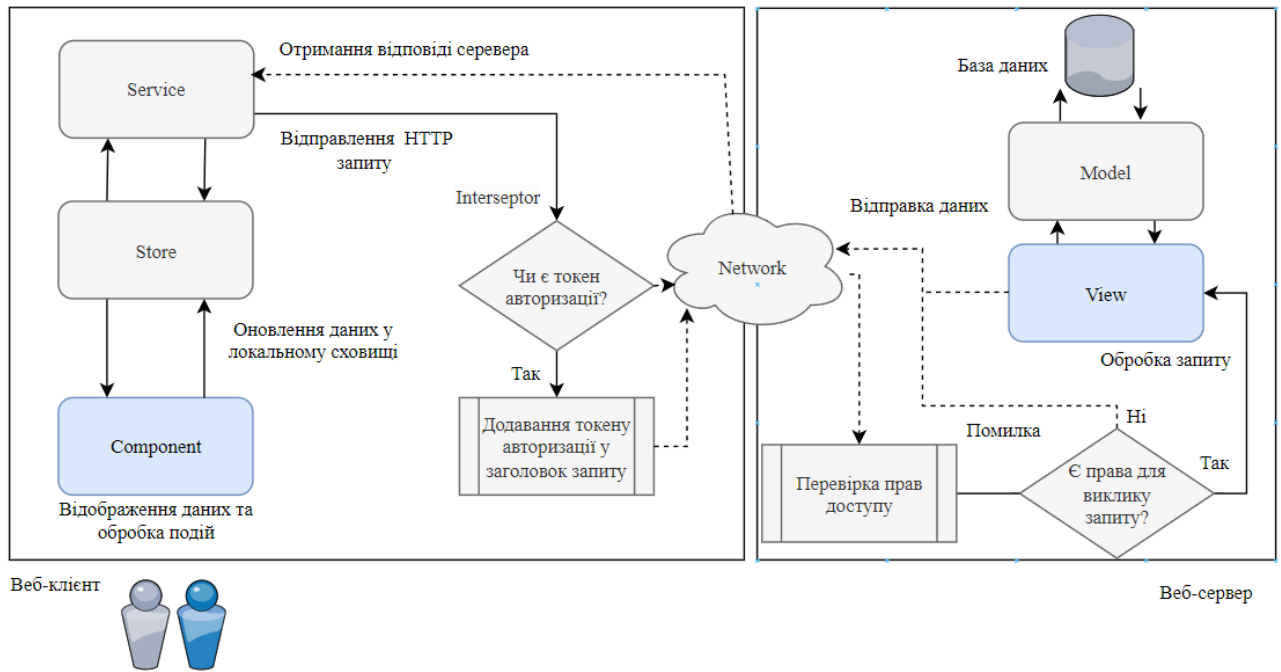


Рис. 3.6. Схема взаємодії компонентів клієнта та сервера

3.1.1. Опис структури сервера

Web-сервер включає три основних модулі: productsServer, auth, products.

Функціональні модулі products та auth містять логіку обробки функціоналу продуктового кошику та авторизації відповідно. Їх основні елементи:

- urls.py – прив'язка API шляхів до відповідних класів обробки запитів;
- views.py – містить класи обробки запитів, виконують перевірки, обробки вхідних даних, виклик допоміжних функцій, методів моделей та повернення результатів до клієнта
- models – містить список моделей систем, що представляють таблиці бази даних у вигляді класів;

– `serializers.py` – містить класи для серіалізації та де серіалізації моделей: перетворення даних у JSON формат та навпаки;

– `admin.py` – налаштування відображення даних у панелі адміністратора;

– `helpers` – містить файли з відокремленням функціоналу розрахунків, конвертації даних у файли та отримання вартості продуктів.

У модулі `products basket main` містяться файли глобальної маршрутизацією та налаштування проекту:

– `urls` – глобальна маршрутизація, поєднання модулів та їх основних маршрутів;

– `settings` – налаштування системи: підключення зовнішніх модулів, налаштування зв'язку з базою даних, локалізація, налаштування точена авторизації та інше (рис. 3.7).

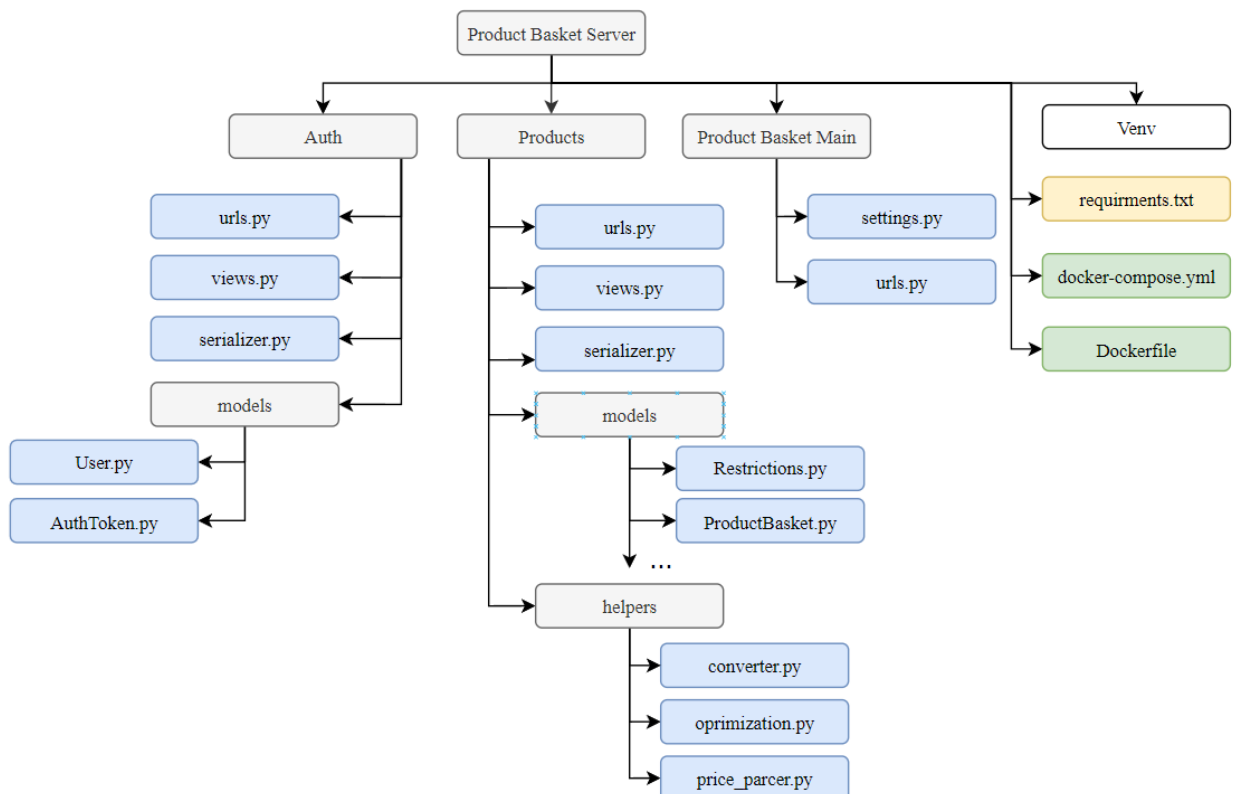


Рис. 3.7. Структура компонентів серверу

У головній теці також міститься:

– `venv` – описує віртуальне оточення проекту та підключені бібліотеки;

- requirements.txt – список підключених бібліотек та їх версій;
- docker-compose.yml – містить інструкції для запуску та налаштування сервісів;
- Dockerfile – містить інформацію для побудови образу.

3.1.2. Опис структури клієнта

Клієнт містить структуру модулів та компонентів, стилів та допоміжних елементів для побудови інтерфейсу клієнта, організації взаємодії з користувачем та сервером.

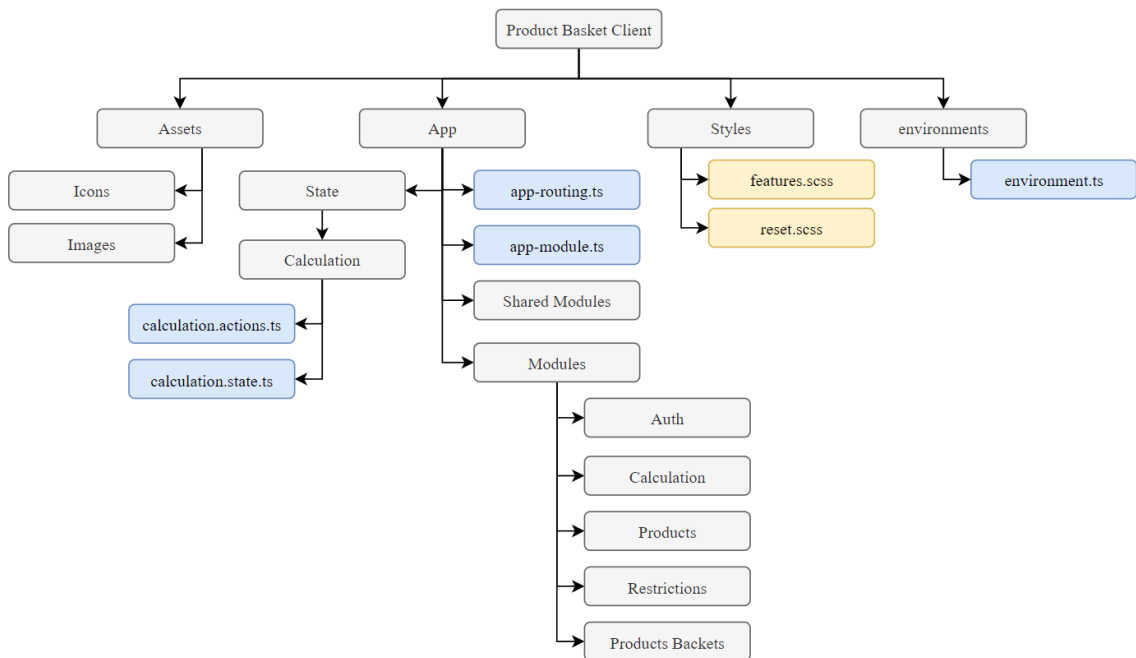


Рис. 3.8. Структура клієнта

Основна тека веб-клієнту містить наступні групи елементів (рис. 3.8):

- app – включає основні компоненти, модулі та ресурси додатку;
- assets – включає основні компоненти, модулі та ресурси додатку;
- styles – включає глобальні стилі проекту відсортовані за призначенням;
- environments – включає опис змінних оточення.

У теці app знаходяться файли з описом глобальної маршрутизації проекту, оголошенням та підключенням зовнішніх модулів.

- `state` – містить опис сховищ даних та подій зміни даних;
- `modules` – включає тематичні модулі, що описують функціонал: авторизації, перегляду та налаштування продуктів, перегляду та налаштування обмежень, списку розрахованих кошиків та оптимізації продуктового кошику;
- `shared-modules` – включає елементи інтерфейсу, що використовуються у різних модулях програми: таблиці, елементи форм, інше.

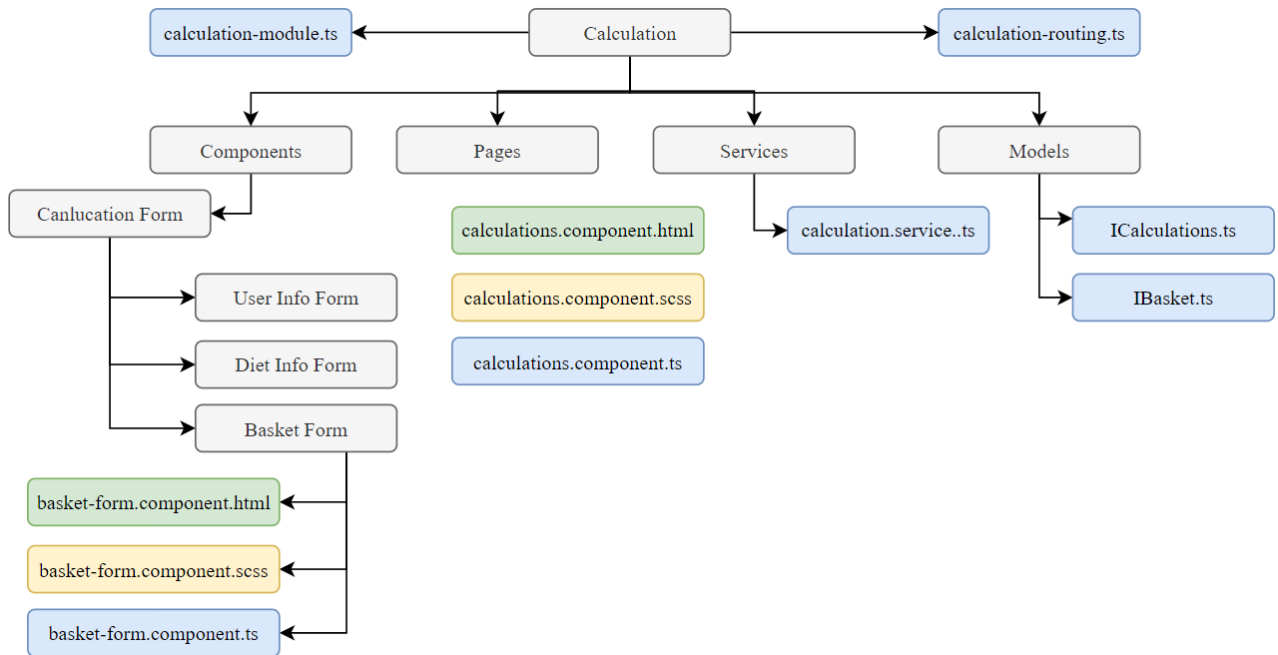


Рис. 3.9. Структура модуля Calculations

Тематичні модулі мають наступну структуру (рис. 3.9):

- `<module-name>.module` – головний клас модуля, опис та підключення зовнішніх модулів;
- `<module-name>-routing.module` – описом маршрутів модуля та зв'язок з компонентами відображення;
- `services` – сервіси з логікою зв'язку з зовнішнім сервером;
- `models` – опис типів, інтерфейсів та класів модуля;
- `components` – містить внутрішні компоненти модуля, такими як форми, таблиці;
- `pages` – містить головні компоненти модуля, прив'язані до маршрутів.

3.1.3. Інструменти розробки та розгортання додатку

При роботі з було використано низьку програмних засобів для організації збереження коду, розробки та розгортання додатку.

Розробка програмного додатку проводилася у інтегрованих середовищах розробки WebStorm та PyCharm. Це функціональні редактори, що спрощують процес навігації, рефакторингу та контролю якості коду, а також надають низьку інструментів для взаємодії з терміналом, базою даних, системою контролю версій [28].

Для зберігання та контролю версій коду було використано Git, система контролю версій, яка була спроектована Лінусом Торвальдсом. Вона дозволяє гнучко перемикатися між різними версіями коду, відкачувати результати та обробляти конфлікти [29], що підвищує ефективність розробки, особливо командної. Завдяки Git можна організувати процес роботи з різними версіями файлу, об'єднання та перевірки змін (рис. 3.10).

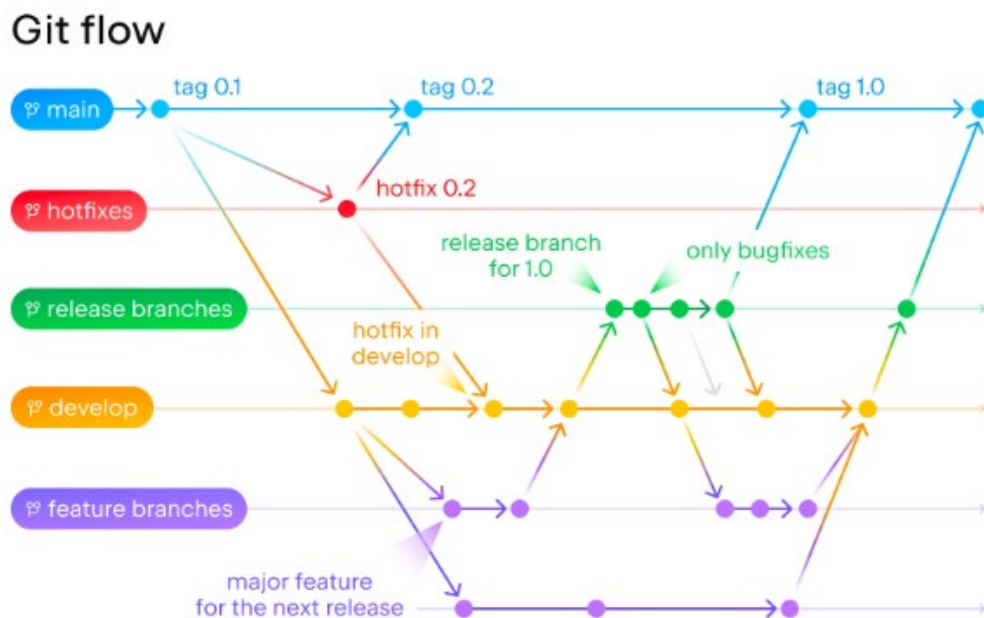


Рис. 3. 10. Процес розробки з використанням системи Git

Для розгортання web-серверу та серверу бази даних було використано Docker, програмне забезпечення яка дозволяє створювати спеціальні контейнери, що відокремлюються від основної частини операційної системи та

працюють як окремі процеси. Таким чином пришвидшуються процес деплою програмних продуктів та створюється незалежність від зовнішнього середовища [30].

3.1.4. Технічні вимоги до розгортання та запуску сервісів

Завдяки особливості мікросервісної архітектури сервери для розгортання основного сервера та клієнта можуть знаходитися як на різних пристроях, так і на одному. Найбільш ефективно буде використання хмарного серверу, що має наступні мінімальні технічні характеристики:

- оперативна пам'ять [RAM]: 512 МБ;
- кількість ядер процесору: 4;
- накопичувач [HDD]: 500 МБ.

Для розгортання веб-серверу у головній директорії проекту необхідно запустити команду для зборки та запуск серверу бази даних та веб-серверу «docker-compose up». Для перевірки роботи програми потрібно відкрити інтерфейс адміністратора за шляхом <http://0.0.0.0:3000/admin>.

Для розгортання веб-клієнта потрібно виконати команду «ng serve». Проект буде запущено за адресою <http://localhost:4300>

3.2. Структура та опис бази даних

При побудові бази даних було виділено головні сутності програмної системи та їх зв'язки. Для збереження персональних даних користувача для розрахунків та авторизації була виділена сутність «Користувач» з полями: ім'я, прізвище, адресу електронної пошти, стать, вік, вагу, зріст, рівень активності.

Сутність «Дієта» зберігає дані про назву, опис дієти, співвідношення макронутрієнтів та обмеження вживання продуктів. Сутність «Продукт» зберігає дані про назву, категорію продукту, його енергетичну цінність та вартість. Сутність «Обмеження» зберігає дані про обмеження вживання продукту: продукт, умову обмеження та кількість продукту.

Сутність «Продуктовий кошик» зберігає дані про розрахований продуктивний кошик: назву, дату створення, енергетичну цінність, загальну вартість та список продуктів.

Для розділення логіки авторизації та даних для розрахунку сутність «Користувач» було розбито на дві таблиці: «Користувач» та «Параметри користувача», які пов'язані зв'язком «один-до-одного». Опис полів дивись у табл. 3.1.-3.2.

Таблиця 3.1

Опис сутностей користувача

Атрибут	Опис
id	integer (унікальний ідентифікатор)
username	string (логін користувача)
password	string (хеш паролю користувача)
email	string (електронна пошта користувача)
is_superuser	boolean (прапорець для визначення адміністратора)

Таблиця 3.2

Опис сутності параметрів користувача

Атрибут	Опис
id	integer (унікальний ідентифікатор)
user_id	integer (посилання на ідентифікатор користувача)
height	integer (зріст користувача)
weight	float (вага користувача)
years	integer (вік користувача)

Кожна дієта може мати має власний набір обмежень, а одне обмеження може відноситися до кількох дієт. Для розриву зв'язку «багато до багатьох»

виділимо окрему проміжну таблицю, що буде містити ідентифікатори дієти та обмеження. Обмеження пов'язана зв'язком «багато до одного» з продуктом: один продукт може мати різні обмеження, а одне обмеження відноситься до одного продукту. Опис полів дивись у табл. 3.3-3.4.

Таблиця 3.3

Опис сутності «Дієта»

Атрибут	Опис
id	integer (унікальний ідентифікатор)
name	string (назва дієти)
description	text (опис дієти)
carbs_min	float (мінімальна кількість вуглеводів)
carbs_max	float (максимальна кількість вуглеводів)
prot_min	float (мінімальна кількість білків)
prot_max	float (максимальна кількість білків)
fat_min	float (мінімальна кількість жирів)
fat_max	float (максимальна кількість жирів)

Таблиця 3.4

Опис сутності «Обмеження»

Атрибут	Опис
id	integer (унікальний ідентифікатор)
product_id	integer (посилання на ідентифікатор продукту)
comparator	string (оператор порівняння: <, >, =)
amount	float (кількість обмеження)
unit	string (одиниця виміру: грам, літрів, штук)
default	boolean (показник обмеження за замовчуванням)

Дієта може мати обмеження на певні групи продуктів. Наприклад, м'ясні вироби у вегетаріанській дієті. Для відображення цих обмежень введемо таблицю «Виключені Категорії Дієти», що містить ідентифікатори дієти та категорії, дивись табл. 3.5.

Таблиця 3.5

Опис сутності «Виключені Категорії Дієти»

Атрибут	Опис
diet_id	integer (посилання на ідентифікатор дієти)
category_id	integer (посилання на ідентифікатор категорії)

Користувач може додавати власні списки продуктів та обмежень у систему, таким чином для збереження цього зв'язку було виділено проміжні таблиці «Продукти користувача» та «Обмеження користувача», дивись табл. 3.6-3.7

Таблиця 3.6

Опис сутності «Продукти користувача»

Атрибут	Опис
user_id	integer (посилання на ідентифікатор користувача)
product_id	integer (посилання на ідентифікатор продукту)

Таблиця 3.7

Опис сутності «Обмеження користувача»

Атрибут	Опис
user_id	integer (посилання на ідентифікатор користувача)
restriction_id	integer (посилання на ідентифікатор обмеження)

Так як один продукт може мати різну енергетичну цінність у різних станах приготування, було виділено окрему сутність «Стан продукту». Також для уникнення дублювання даних поле категорія була перенесена у окрему таблицю «Категорія» та пов'язана з продуктом зв'язком «багато до одного», дивись табл. 3.8-3.10.

Таблиця 3.8

Опис сутності категорія

Атрибут	Опис
id	integer (унікальний ідентифікатор)
name	string (назва категорії)
description	text (опис категорії)

Таблиця 3.9

Опис сутності продукт

Атрибут	Опис
id	integer (унікальний ідентифікатор)
category_id	integer (посилання на ідентифікатор категорії)
name	string (назва продукту)

Таблиця 3.10

Опис сутності «Стан продукту»

Атрибут	Опис
id	integer (унікальний ідентифікатор)
state	string (стан продукту: готовий, варений, жарений, печений)
dish_name	string (назва страви або продукту)
energy	float (кількість енергії в продукті)

1	2
carbohydrates	float (кількість вуглеводів в продукті)
proteins	float (кількість білків в продукті)
fats	float (кількість жирів в продукті)

Також продукт може мати різну вартість в залежності від магазину, тому було інформація про вартість була перенесена у таблицю «Продукт у магазині», яка пов'язана з «Продуктом» зв'язком «багато до одного». Опис полів дивись у табл. 3.11.

Таблиця 3.11

Опис сутності «Продукт у магазині»

Атрибут	Опис
id	integer (унікальний ідентифікатор)
name	string (назва продукту в магазині)
price	decimal (ціна продукту)
amount	integer (кількість доступних одиниць)
unit	string (одиниця виміру: грам, літрів, штук)
id	integer (унікальний ідентифікатор)
name	string (назва продукту в магазині)

Сутності «Продуктовий кошик» зберігає інформацію про проведені розрахунки користувача для майбутнього перегляду та завантаження. Так як набір та вартість продуктів змінюються з часом, тут зберігається зріз обраного списку продуктів у форматі JSON.

Опис всіх полів дивись у табл. 3.12.

Опис сутності «Продуктовий кошик»

Атрибут	Опис
user_id	integer (посилання на ідентифікатор користувача)
id	integer (унікальний ідентифікатор кошика)
name	string (назва кошика)
creation_date	date (дата створення кошика)
period	integer (період використання кошика, кількість днів)
sum	float (загальна сума кошика)
products	json (дані про продукти у кошику у форматі json)

За побудованими таблицями та зв'язками, була створена ER-діаграма бази даних представлена на рис. 3.11. Діаграма включає також автоматично створені системою таблиці для збереження проміжних зв'язків та даних авторизації.

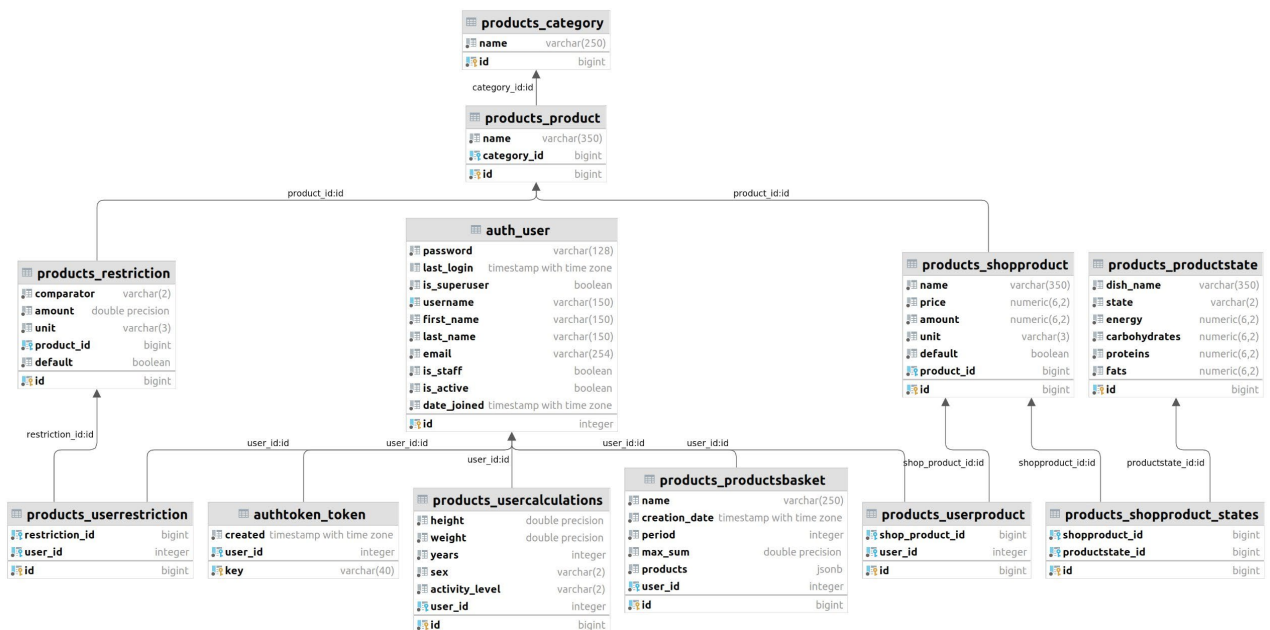


Рис. 3.11. ER-діаграма бази даних

Кожній описаній сутності відповідає клас програмної системи. Приклади опису класів представлені у додатку А.

3.3. Розробка API інтерфейсу сервера

При проектуванні та розробці API веб-сервера було виділено дві групи запитів: запити, що відносяться до профіля користувача, авторизації та реєстрації та запити, що відносяться до продуктового кошика та пов'язаних з ним даних.

Нижче представлено опис запитів, що відносяться до групи API продуктового кошика.

Ініціалізація процесу оптимізації *products/optimization*. Приймає наступні вхідні параметри:

- dietId – ідентифікатор дієти;
- kkalAmount – необхідна кількість енергії на добу;
- term – кількість діб для виконання розрахунку;
- maxSum – максимальне обмеження бюджету;
- categoriesToExclud – список категорій продукту для виключення.

Результатом виконання запиту є оптимізований список продуктів, їх загальна вартість та енергетична цінність.

Додавання продукту *products/products*. Приймає дані про продукт:

- name – назва продукту;
- energyAmount – кількість калорій на 100 г продукту;
- prots – кількість білків на 100 г продукту;
- carbs – кількість вуглеводів на 100 г продукту;
- fats – кількість вуглеводів на 100 г продукту;
- price – вартість продукту;
- amount – кількість продукту;
- unit – одиниця продукту.

Результатом виконання запиту є створення записів у таблицях «Продукт», «Продукт у магазині» та «Стан продукту» і встановлення зв'язків між ними.

Додавання обмеження щодо споживання *products/restrictions*. Приймає наступні вхідні дані:

- *productId* – ідентифікатор продукту;
- *comparator* – умова обмеження;
- *amount* – кількість продукту;
- *unit* – одиниця продукту.

Результатом виконання запиту є створення нового запиту у таблиці «Обмеження» з додаванням зв'язка до користувача, що створив обмеження.

Для реалізації функціоналу отримання даних продуктового кошику було розроблено наступні запити:

- *products/optimization* – отримання списку продуктових кошиків користувача;
- *products/product-basket/<int:id>* – отримання одного продуктового кошику користувача за ідентифікатором кошику;
- *products/categories* – отримання списку категорій продуктів;
- *products/products* – отримання списку продуктів;
- *products/restrictions* – отримання списку обмежень;
- *products/diets* – отримання списку дієт.

Для реалізації функціоналу редагування даних було розроблено наступні запити:

- *products/products/<int:id>* – редагування продукту за його ідентифікатором;
- *products/restrictions/<int:id>* – редагування обмеження за його ідентифікатором.

Для реалізації функціоналу видалення даних було розроблено наступні запити:

- *products/product-basket/<int:id>* – видалення продуктового кошику за його ідентифікатором;
- *products/products/<int:id>* – видалення продукту за його ідентифікатором;
- *products/restrictions/<int:id>* – видалення обмеження за його ідентифікатором.

Нижче представлено опис запитів, що відносяться до групи API авторизації та реєстрації.

Запит *auth/token/* для реєстрації входу у систему та отримання токена авторизації приймає наступні вхідні параметри:

- *username* – ім'я користувача;
- *password* – пароль користувача.

Запит *auth/token/refresh/* викликається для оновлення токена авторизації та продовження сесії користувача.

Запит *auth/profile/create/* для створення профілю користувача приймає наступні вхідні дані:

- *username* – ім'я користувача;
- *password* – пароль користувача;
- *age* – вік користувача;
- *weight* – вага користувача;
- *height* – зріст користувача;
- *activityLevel* – рівень фізичної активності користувача.

Запит *auth/profile* виконує отримання профілю користувача за переданим токеном авторизації.

Для документування доступу та тестуванню API запитів було підключено бібліотеку Swagger, що дозволяє створювати інтерактивну документацію до запитів серверу [31].

GET	/products/categories	products_categories_list
GET	/products/diets	products_diets_list
GET	/products/optimization	products_optimization_list
POST	/products/optimization	products_optimization_create

Рис. 3.12. Перегляд списку запитів

Swagger дозволяє переглядати список запитів (рис. 3.12), переглядати приклади відповідей (рис.3.13) та прикладі даних для запиту, а також тестувати виконання запитів з передачею параметрів через Swagger-інтерфейс (рис. 3.14).

Code	Description
200	<div style="display: flex; justify-content: space-between;"> Example Value Model </div> <pre> Restrictions { id: integer title: ID readOnly: true product: Product { id: integer title: ID readOnly: true name*: string title: Name maxLength: 350 minLength: 1 category: Category { id: integer title: ID readOnly: true name*: string } } comparator*: string title: Comparator Enum: [EQ, GT, LT] amount*: number title: Amount unit*: string title: Unit maxLength: 3 minLength: 1 } </pre>

Рис. 3.13. Прикладу відповіді серверу

Name	Description
data* required object (body)	<div style="display: flex; justify-content: space-between;"> Edit Value Model </div> <pre> { "product": { "name": "string", "category": { "name": "string" } }, "comparator": "EQ", "amount": 0, "unit": "str" } </pre> <div style="text-align: center; margin-top: 10px;"> <input type="button" value="Cancel"/> </div> <div style="margin-top: 10px;"> Parameter content type <input type="text" value="application/json"/> </div> <div style="text-align: center; margin-top: 10px; background-color: #007bff; color: white; padding: 5px; width: 100px; margin-left: auto;"> Execute </div>

Рис. 3.14. Прикладу формування даних для відправки

3.4. Опис програмного продукту

Опис інтерфейсу додатку було розподілено на наступні сценарії: розрахунок продуктового кошику користувача та адміністрування додатку.

Розрахунок продуктового кошику користувача

Головне вікно сайту містить коротку інформацію про призначення додатку, його мету та попередження, щодо використання. У пунктах меню можна перейти до профілю користувача, сторінці розрахунків продуктового кошику та сторінок налаштування списків продуктів та обмежень (рис. 3.15).

Також додаток пропонує два варіанти оформлення сторінок «світлий», що відображається за замовчування там «темний».

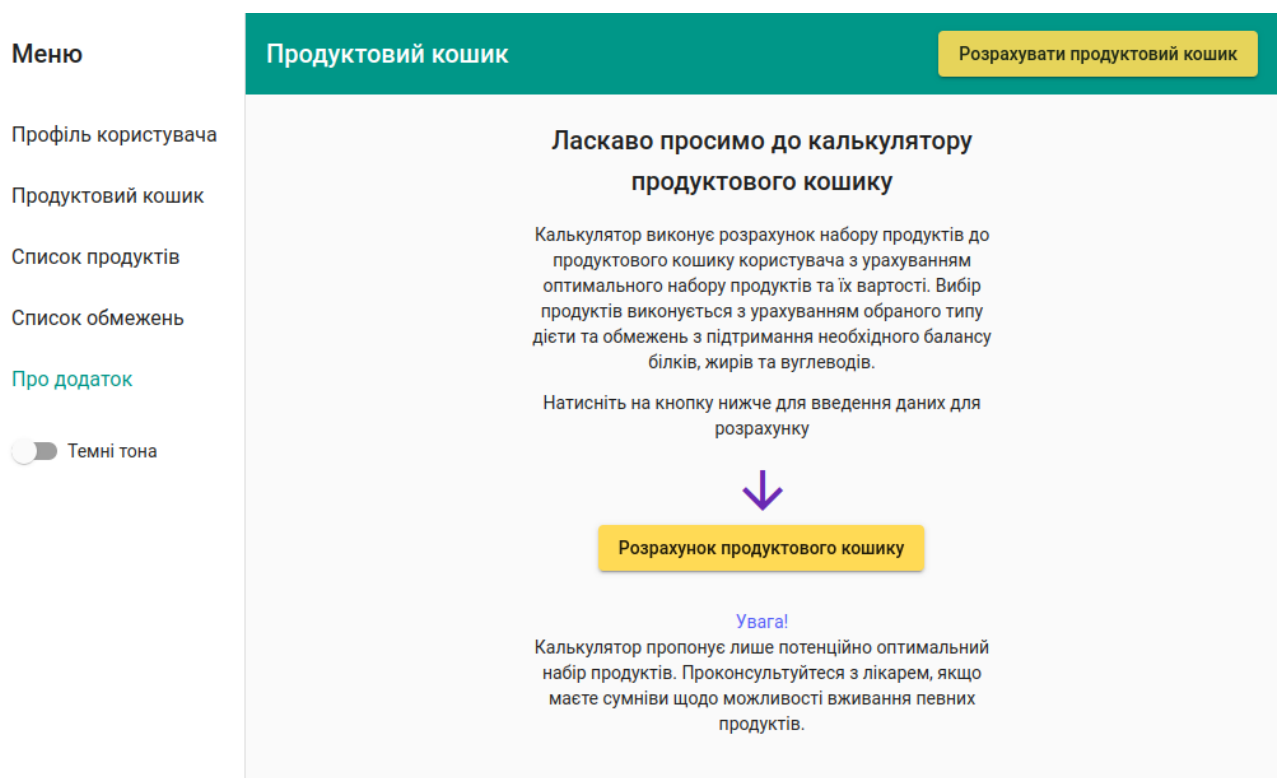


Рис. 3.15. Головна сторінка додатку

Натискання кнопки «Розрахунок продуктового кошику» або пункту меню «Продуктовий кошик» відкриє сторінку, де користувач може переглянути та за необхідністю актуалізувати дані щодо параметрів та рівню фізичної активності, дивись рис. 3.16.

Рис. 3.16. Форма для заповнення персональних характеристик користувача для розрахунку добової норми калорій

Рис. 3.17. Зміна теми та розміру екрана

Натискання кнопки «Далі» або пункту «Напрямок дієти» відкриває наступну форму, де обирається напрямок дієти: низьковуглеводна, високобілкова, дієта для контролю артеріальної гіпертензії, дієта при цукровому діабеті другого типу, середземноморська, вегетаріанська (рис. 3.17). При виборі дієти відкривається її короткий опис, а також відмічаються групи продуктів, що виключаються з розрахунків для даного напрямку харчування. Наприклад, при виборі вегетаріанської дієти з розрахунків автоматично прибирається група м'ясних продуктів.

Користувач також має можливість сам виключити непотрібні йому групи продуктів, як показано на рис. 3.18.

Дані користувача — 2 Напрямок дієти — 3 Термін та вартість — 4 Done

Оберіть тип дієти для розрахунку

Збалансована дієта

Низьковуглеводна дієта

Високобілкова дієта
Сприяє росту та відновленню м'язів, а також підтримці ситості. Може бути ефективною при фізичних тренуваннях та роботі над фігурою.

Дієта для контролю артеріальної гіпертензії

Дієта при цукровому діабеті другого типу

Середземноморська дієта

Вегетаріанська дієта

Прибрати продукти з категорії:

Бакалія ~~Консерви~~ М'ясні продукти Молочні продукти Напої Риба та морепродукти

~~Соуси та спеції~~ Фрукти та овочі Хлібо-булочні продукти Яйця Бобові

Переглянути список продуктів для розрахунків

Рис. 3.18. Вибір типу дієти та виключення категорій продуктів

При натисканні на кнопку «Переглянути список продуктів для розрахунків» буде відкрите вікно з наборами продуктів, згрупованих по категоріях (рис. 3.19).

Користувач має можливість переглядати вартість та енергетичну цінність продуктів, додавати та редагувати власні продукти, а також шукати продукти за назвою.

Продуктовий кошик

Список продуктів для розрахунку

Натисніть на рядок для отримання детальної інформації щодо калорійності та пищевої цінності продукту

Для редагування кількості та вартості продукту наведіть на рядок та натисніть на іконку ✎

Для виключення продукту зі списку натисніть на іконку ✖

Пошук за назвою продукту

Додати продукт

Бакалія

Бобові

Консерви

М'ясні продукти

Молочні продукти

Напої

Рис. 3.19. Перегляд списку продуктів, що буде приймати участь у розрахунках

Для додавання власного продукту потрібно натиснути на кнопку «Додати продукт» та заповнити інформацію щодо назви, категорії, вартості та енергетичних характеристик: кількість енергії, білків, жирів та вуглеводів на 100 грам продукту. Користувач може обрати тип приготування продукту, наприклад «Варений», «Запечений», так як від типу приготування може змінитися і енергетична цінність продукту (рис. 3.20).

При заповненні інформації про вартість продукту потрібно ввести його кількість та обрати одиницю вимірювання: грами, кілограми, літри, міліграми, штуки.

Додати продукт

Введіть дані для додавання нового продукту у список для проведення розрахунків

Категорія продукту *
Фрукти та овочі

Назва продукту *
Авокадо

Вага / кількість продукту *
1

Одиниця вимірювання *
Кілограми (кг)

Вартість продукту *
€ 99

Введіть дані щодо енергетичних характеристик продукту (на 100 гр продукту)

Тип приготування *
-

Калорійність на 100 гр продукту *
159

Білки (гр) *
1.6

Жири (гр) *
13.1

Вуглеводи (гр) *
6

Відмінити

Додати продукт

Рис. 3.20. Приклад додавання продукту користувачем

При натисканні на певну категорію буде відкрито список продуктів цієї категорії з інформацією про їх кількість та вартість (рис.3.21).









Фрукти та овочі			
Назва продукту	Кількість / Вага	Вартість	
Капуста Пекінська	1 кг	17.00 грн.	 
Горіх волоський бланшований	120 гр	84.99 грн.	 
Авокадо	1 кг	99.00 грн.	 
Картопля	1 кг	14.90 грн.	 

Рис. 3.21. Перегляд списків продуктів у категорії «Фрукти та овочі»

При натисканні на продукт буде відкрито інформацію про його енергетичну цінність та вміст основних нутрієнтів у 100г продукту (рис. 3.22-3.23).

Назва продукту		Кількість / Вага	Вартість		
Капуста Пекінська		1 кг	17.00 грн.		
Горіх волоський		120 гр	84.99 грн.		
Стан	Калорійність на 100 г продукту	Білки	Жири	Вуглеводи	
-	718 ккал	15.76 г	63.38 г	18.45 г	

Рис. 3.22. Перегляд інформації про енергетичну цінність волоського горіху

Курятина філе		1 кг	159.30 грн.		
Стан	Калорійність на 100 г продукту	Білки	Жири	Вуглеводи	
Варений	151 ккал	29 г	3 г	0 г	

Рис. 3.23. Перегляд інформації про енергетичну цінність продукту вареного курячого філе

Користувач може видалити продукт, натиснувши на символ кошику у рядку інформації про продукт, та підтвердити видалення (рис 3.24).

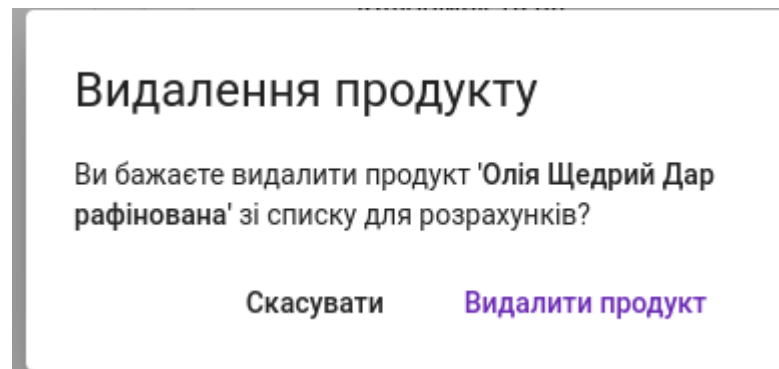


Рис. 3.24. Видалення продукту

При переході на сторінку «Список обмежень» буде відкритий список обмежень за замовчуванням та введених користувачем. Обмеження включає інформацію про продукт, умову обмеження: «більше», «менше», «рівно», кількість продукту та одиницю вимірювання (рис. 2.25).

Наприклад, якщо користувач хоче з'їдати не менш ніж 200 г яблук на добу він може ввести обмеження: «Яблуко, більше 200 г» і це буде взято до уваги під час процесу оптимізації.

Продуктовий кошик

Список обмежень на вживання продукту на добу

Продукт * Червона квасоля	Умова * Менше ніж	Вага / кількість продукту * 200	Одиниця вимірювання * Грами (гр)
------------------------------	----------------------	------------------------------------	-------------------------------------

Додати обмеження

Існуючі обмеження

Продукт	Обмеження	Значення
Яйця курячі	Менше ніж	2 шт
Салат	Рівно	100 гр
Хліб житній	Менше ніж	250 гр

Рис. 3.25. Приклад додавання обмеження

Також користувач може видалити існуюче обмеження, як показано на рисунку 3.26.

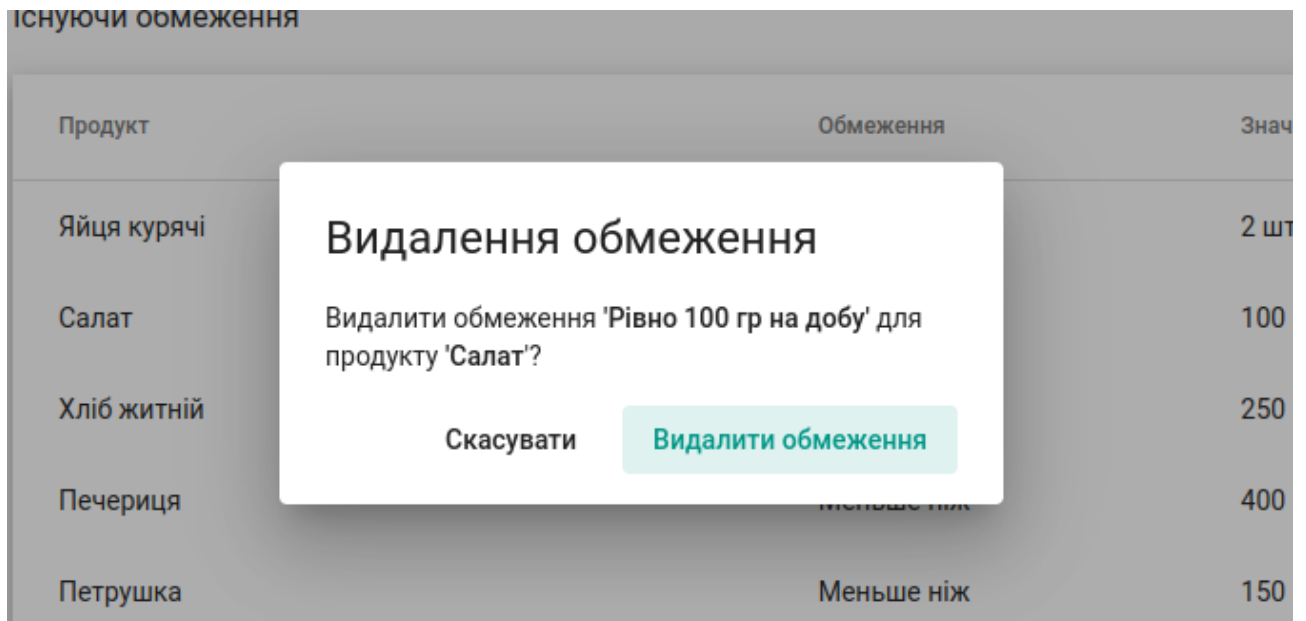


Рис. 3.26. Підтвердження видалення обмеження

При поверненні на сторінку розрахунків на натисканні кнопки «Далі», буде відкрита форма, де заповнюється інформація про термін, на який потрібно заповнити продуктовий кошик, а також вводиться максимальний бюджет на розрахунки (рис. 3.27).

Рис. 3.27. Заповнення інформації щодо терміну та обмеження бюджету

Остання сторінка показує підсумкову інформацію про енергетичні потреби та обмеження бюджету, згідно яких будуть проведені

розрахунки (рис.3.28). Натискання кнопки «Розрахувати продуктивий кошик» запускає процес оптимізації.

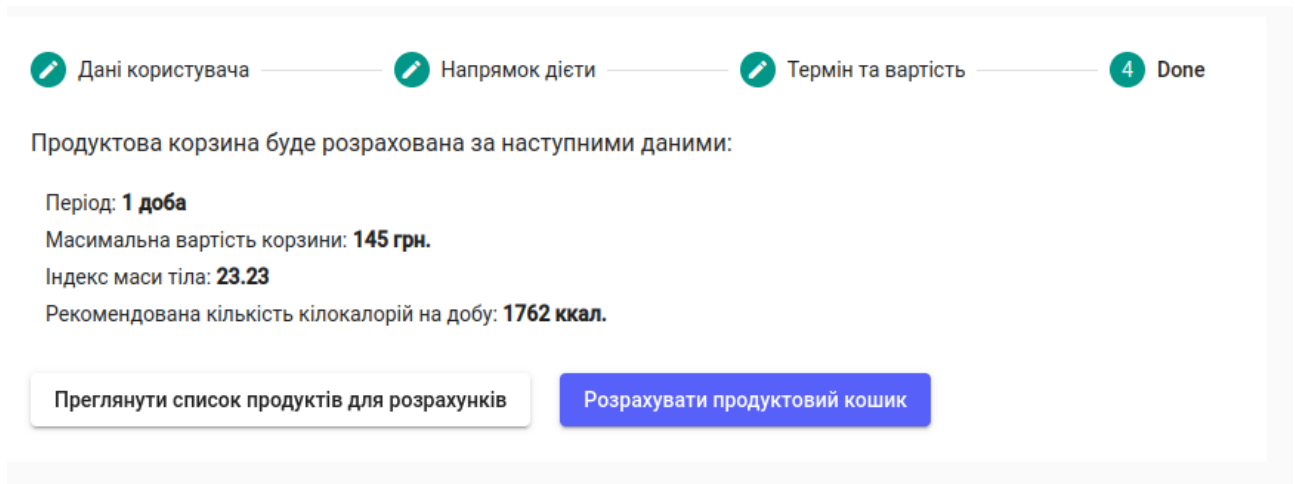


Рис. 3.28. Перегляд інформації про дані за якими будуть проведені розрахунки

Процес оптимізації займає від 3 до 7 секунд та потім користувачу відкриваються результати розрахунків (рис. 3.29).

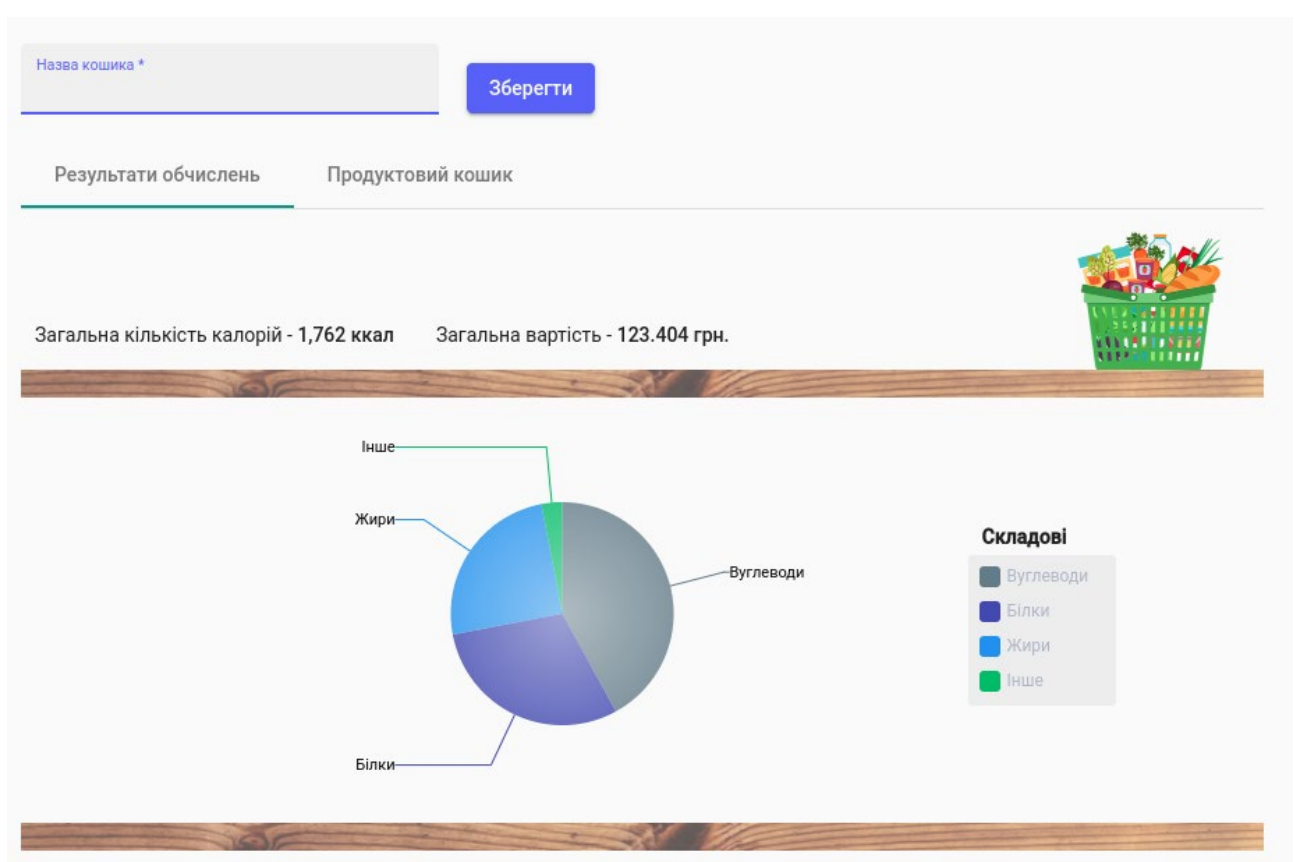


Рис. 3.29. Результати розрахунків

Перша вкладка показує загальні результати, такі як калорійність кошика, загальну вартість та співвідношення білків, жирів, вуглеводів у вигляді кругової діаграми. При переході на наступну вкладку «Продуктовий кошик» відкривається список продуктів, з інформацією про кількість, вартість та енергетичну цінність кожного продукту (рис. 3.30).

Користувач може задати назву продуктового кошика та зберегти зміни.

Результати обчислень		Продуктовий кошик	
Назва продукту		Кількість / Вага	Вартість
Йогурт натуральний несолодкий (біо)		100 гр	7.02 грн.
Горіх волоський бланшований		28 гр	19.83 грн.
Горіхова каша		200 гр	3.33 грн.
Пластівці вівсяні		175 гр	5.61 грн.
Яйця курячі Ясенвіт (С1 10 шт)		1 шт	5.99 грн.
Салат Пучок-Свіжачок		100 гр	37.94 грн.
Морква		20 гр	0.20 грн.
Перець болгарський жовтий		10 гр	1.38 грн.
Яблука українське		100 гр	1.71 грн.
Курятина гомілка		200 гр	17.39 грн.
Курятина філе		138 гр	21.97 грн.
Гречка		30 гр	0.79 грн.
Олія Щедрий Дар рафінована		4 гр	0.24 грн.

Рис. 3.30. Список продуктів

При натисканні на продукту відкривається детальна інформація про його енергетичну цінність та кількість нутрієнтів у представленій кількості продукту (рис. 3.31).

Назва продукту		Кількість / Вага		Вартість	
Йогурт натуральний несолодкий (біо)		100 гр		7.02 грн.	
Стан	Калорійність	Білки	Жири	Вуглеводи	
	56 ккал	4 г	2 г	5 г	
Горіх волоський бланшований		28 гр		19.83 грн.	
Горіхова каша		200 гр		3.33 грн.	

Рис. 3.31. Перегляд інформації про продукт та його енергетичні характеристики

Користувач може роздрукувати результати або завантажити у вигляді файлу, як показано на рис. 3.32.

	A	B	C	D	E	F	G
1	Назва продукту	Кількість (г)	Калорійність, ккал	Вартість, грн	Жири, г	Вуглеводи, г	Білки, г
2	Молоко	150,00	18,00	0,75	1,05	1,50	4,17
3	Горіхова каша	200,00	194,00	14,00	1,18	34,00	3,33
4	Пластівці вівсяні	200,00	800,00	26,28	13,74	136,18	6,40
5	Яйця курячі Ясенсвіт (С1 10 шт)	100,00	1,60	0,12	0,10	0,01	5,98
6	Хліб житній Український Столичний	130,00	313,30	9,75	1,30	62,79	4,47
7	Капуста білокачанна	28,00	8,69	0,56	0,07	1,12	0,31
8	Бурак	14,00	6,50	0,29	0,01	1,44	0,15
9	Яблука українське	100,00	63,00	0,37	0,40	13,00	1,71
10	Бекон Ятрань іспанський	38,00	183,40	3,75	18,75	0,00	9,43
11	Курятина гомілка	159,00	189,30	33,41	5,41	0,00	13,83
12	Гречка	33,00	34,79	0,99	0,21	6,63	0,87
13	Олія Щедрий Дар рафінована	13,00	116,61	0,00	13,00	0,00	0,86
14	Цукор кристалічний	19,00	76,81	0,00	0,00	19,15	0,61
15	Загальні значення		2006,00	52,15	55,22	275,82	90,27
16							

Рис. 3.32. Приклад завантажених результатів у вигляді ексел файлу

На сторінці «Профіль користувача» відображається загальна інформація про поточного користувача та його розраховані раніше продуктові кошики (рис. 3.33).

- Меню
- Профіль користувача
- Продуктовий кошик
- Список продуктів
- Список обмежень
- Про додаток
- Темні тона

Продуктовий кошик

Розрахувати продуктовий кошик

Кабінет користувача Анна:

Індекс маси тіла: **23.05**
Рекомендована кількість кілокалорій на добу: **1951 ккал.**

Продуктові кошики

Назва	Період	Максимальна сума	Дата створення
Список за 11/12/2023 19:28:05	1	120 грн	12/11/23, 21:28
Список за 10/12/2023 13:20:01	1	120 грн	12/10/23, 15:20
Список за 09/12/2023 20:48:22	1	120 грн	12/9/23, 22:48

Рис. 3.33. Кабінет користувача зі списком попередніх розрахунків

Користувач може розраховувати кошик на різні терміни, наприклад, на тиждень, для планування харчування та бюджету (рис. 3.34).

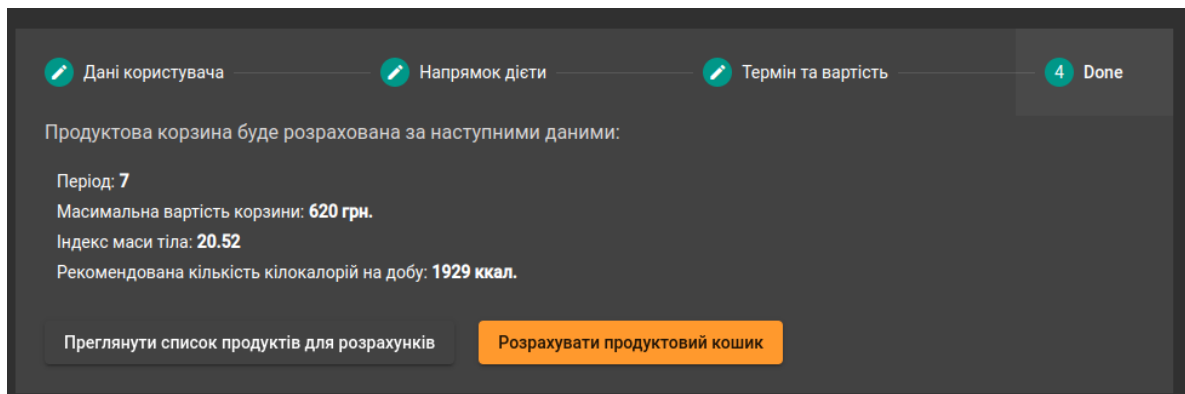


Рис. 3.34. Параметри для розрахунку кошику на тиждень

За результатами оптимізації буде сформовано продуктивий кошик на тиждень, що буде мати більш різноманітний список продуктів. Такий тип розрахунків зручний для широкого планування закупівель (рис. 3.35-3.36)

Назва продукту	Кількість / Вага	Вартість
Молоко топлене	287 гр	17.23 грн.
Горіх волоський бланшований	196 гр	138.82 грн.
Квасоля сушена кольорова	350 гр	30.09 грн.
Пластівці вівсяні	800 гр	25.59 грн.
Хліб житній Український Столичний	162 гр	5.58 грн.
Хліб пшеничний	129 гр	4.49 грн.
Капуста білокачанна	700 гр	7.86 грн.
Перець болгарський жовтий	70 гр	9.65 грн.
Яблука українское	700 гр	11.98 грн.
Кефір Селянський 2,5%	1400 гр	63.27 грн.
Пшоно	659 гр	22.09 грн.

Рис. 3.35. Приклад частини списку продуктів з розрахунків на тиждень

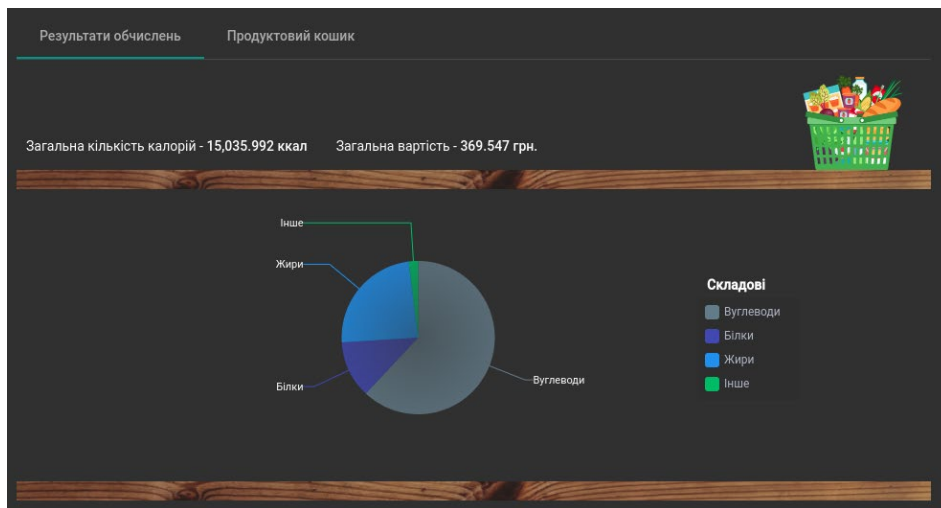


Рис. 3.36. Приклад результатів розрахунків на тиждень

Опис інтерфейсу адміністратора та менеджера

Кабінет адміністратора знаходиться за шляхом 0.0.0.0:8000/admin. При переході за посилання з'являється форма для авторизації (рис. 3.36) з перевіркою введених даних та виводу попередження при некоректному паролі чи імені користувача.

The screenshot shows a login form for an administrator. It has a white background with a blue header bar. The form contains two input fields: 'Username:' with the text 'west' and 'Password:' with masked characters '.....'. Below the fields is a blue 'Log in' button.

Рис. 3.37. Форма авторизації адміністратора

Адміністратор може переглянути списки користувачів системи (рис. 3.38), шукати та відфільтрувати за роллю, а також видалити, заблокувати користувача за потреби.

Select user to change ADD USER +

Q Search

Action: Go 0 of 11 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Ajw	nastia.kyrakina+3@gmail.com			✖
<input type="checkbox"/>	Anastasia	anna.last@gmail.com			✖
<input type="checkbox"/>	Anna	nastia.kyrakina@gmail.com			✖

FILTER

By staff status

All
Yes
No

By superuser status

All
Yes
No

Рис. 3.38. Сторінка адміністрування

Сторінка менеджера містить у собі інформацію про збережені у системі набори продуктів, діет, обмежень та історію змін (рис. 3.39).

Site administration

PRODUCTS	
Categories	+ Add Change
Diet category restrictions	+ Add Change
Diet product restrictions	+ Add Change
Diets	+ Add Change
Product states	+ Add Change
Products	+ Add Change
Products baskets	+ Add Change
Restrictions	+ Add Change
Shop products	+ Add Change
User calculationss	+ Add Change

Recent actions

My actions

- ✖ Молоко Простоквашино 1% / 32.90
Shop product
- ✖ Молоко Яготинське 2,6% / 40.39
Shop product
- + Дієта для контролю артеріальної гіпертензії Молоко GT 150.0
Diet product restriction
- + Молоко GT 150.0
Restriction
- ✖ Кефір / 30.00
Shop product
- ✖ Кефір біфідок / 35.10

Рис. 3.39. Перегляд історії дій та списків даних системи

Для додавання нових даних, наприклад обмеження потрібно зайти на відповідну вкладку, натиснути на кнопку «Add restriction» та ввести дані обмеження (рис. 3.40-3.41).



Add diet product restriction

Diet:

Restriction:

Рис. 3.40. Додавання обмеження для дієти

Add restriction

Product: Червона квасоля  

Comparator: Less than

Amount: 200

Unit: гр

Default

Save and add another Save and continue editing SAVE

Рис. 3.41. Додавання обмеження за замовчуванням

Менеджер має можливість переглядати та контролювати списки продуктів, категорій, дієт (рис. 3.42).

Select shop product to change ADD SHOP PRODUCT +

Action: Go 0 of 100 selected

<input type="checkbox"/>	SHOP PRODUCT
<input type="checkbox"/>	Авокадо / 170.00
<input type="checkbox"/>	Оливкова олія / 160.00
<input type="checkbox"/>	Пластівці вівсяні / 12.80
<input type="checkbox"/>	Горіхова каша / 14.99
<input type="checkbox"/>	Квасоля сушена кольорова / 42.99
<input type="checkbox"/>	Горіх волоський бланшований / 84.99
<input type="checkbox"/>	Цукор кристалічний / 32.10

Рис. 3.42. Перегляд списку продуктів у магазині

3.5. Аналіз ефективності моделювання набору продуктів

Для перевірки ефективності роботи програмного додатку було виконано порівняння результатів алгоритму оптимізації з рекомендованими планами харчування відповідних дієт.

Дані щодо вартості продуктів харчування були отримані з онлайн ресурсу index.minfin.com.ua [17], а також з сайтів супермаркетів Сільпо, АТБ, Варус. Інформація щодо енергетичної цінності продуктів були отримані з веб-додатку «Таблиця калорійності» [32].

Метою аналізу є перевірка коректності обчислень програми оптимізації: відповідність вимогам дієти, різноманіття продуктів, порівняння наборів продуктів та значень вартості та енергетичної цінності.

Так як страви кожної дієти є різноманітними для порівняння були зібрані найбільш типові приклади раціонів, що містять представників всіх рекомендованих груп продуктів.

Для аналізу було обрано наступні дієти: дієта для контролю артеріальної гіпертензії, також відома як DASH-дієта, високобілкова дієта та вегетаріанська.

Високобілкова дієта передбачає споживання 25-30% й більше білків у щоденному раціоні. Вміст вуглеводів та жирів коливається у межах 40%-45% та 25-30% відповідно. Рекомендовано різноманіття страв з білками рослинного та тваринного походження. Приклад добового раціону даної дієти був сформований згідно планів харчування з ресурсів [33-34], дивись у табл. 3.13.

Таблиця 3.13

Приклад добового раціону високобілкової дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Сир кисломолочний	200,00	46,58	240,00	34,00	10,00	3,60
Запечена куряча грудка	454,00	71,21	653,17	141,52	7,71	4,08
Чорний хліб	100,00	3,40	201,00	6,60	1,10	41,00
Йогурт	150,00	13,35	160,50	6,00	12,00	7,05
Запечена риба	150,00	25,05	154,50	10,20	7,05	12,60
Салат овочевий	150,00	49,50	85,50	3,00	4,50	9,00
Яйця	150,00	0,18	2,37	0,36	0,33	0,03
Оливкова олія	17,20	17,16	154,11	0,01	17,10	0,03
Броколі	150,00	11,85	43,50	4,05	0,75	3,60
Цибуля шаллот	30,00	11,97	21,60	0,75	0,00	5,10
Сир твердий	75,00	34,87	260,25	16,88	21,38	3,52
Варений бурий рис	150,00	8,55	166,50	3,90	1,20	36,30
Родзинки	83,00	12,36	217,46	1,49	0,00	58,93
Сумарні значення		209,09	2 360,46	37,74%	31,69%	31,32%

Дана дієта включає велику кількість м'ясних та рибних продуктів, через що її середня вартість більша за вартість звичайного раціону харчування.

Для створення продуктового кошика було виконано моделювання дієти для чоловіка 30 років, зростом 170 та вагою 67 кг, що має високий рівень фізичної активності та займається активними тренуваннями від 6 разів на тиждень. Для створення дефіциту калорій рекомендовану енергетичну цінність добового раціону було зменшено на 20%. Згідно введених даних був сформований продуктовий кошик, дивись табл. 3.14.

Таблиця 3.14

Результат оптимізації продуктового кошика для високобілкової дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Йогурт натуральний	100,00	7,02	56,00	4,00	2,00	5,00
Горіх волоський	28,00	19,83	201,04	4,41	17,75	5,17
Горіхова каша	200,00	3,33	194,00	14,00	1,18	34,00
Пластівці вівсяні	200,00	6,40	800,00	26,28	13,74	136,18
Яйця курячі	1,00	5,99	1,60	0,12	0,10	0,01
Хліб житній	58,00	1,98	138,94	4,32	0,58	27,85
Салат	100,00	37,94	25,00	3,00	0,70	2,00
Морква	20,00	0,20	7,00	0,20	0,04	1,40
Перець болгарський	10,00	1,38	2,70	0,10	0,00	0,50
Яблуко українське	100,00	1,71	63,00	0,37	0,40	13,00
Курятина стегно	33,00	3,70	39,11	6,57	1,28	0,00
Курятина гомілка	200,00	17,39	238,00	42,00	6,80	0,00
Курятина філе	200,00	31,86	302,00	58,00	6,00	0,00
Гречка	30,00	0,79	31,50	0,90	0,19	6,00
Олія рафінована	10,00	0,67	90,50	0,00	10,09	0,00
Сумарні значення		140,19	2190,40	30%	25%	42%

У обох таблицях представлено різноманіття продуктів: м'яса, каш, овочів, фруктів, олій. Результати оптимізації пропонують більшу кількість каш, які багаті на білки та вуглеводи. Так як метою високобілкової дієти є нарощення м'язової маси багато рекомендованих для неї дієт мають перевищені за рекомендовані показники кількості білків, що ефективно для короткого терміну харчування, але небезпечно для довготривалого. Програма оптимізації намагається тримати цю кількість у межах норми.

За результатами розрахунку можна побачити, що алгоритм підбирає максимальну кількість більш дешевих продуктів та калорійних продуктів з високим вмістом білків, таких як м'ясо курки та вівсяні пластівці. Таким чином потерпає різноманіття дієти, але суттєво зменшуються витрати на закупівлю.

Дієта для контролю артеріальної гіпертензії фокусується на обмеженні споживанні продуктів з великим вмістом жирів та складних вуглеводів, надаючи перевагу фруктам, овочам, кашам та білому м'ясу. Рекомендована кількість вуглеводів на день складає 50-55% від добового раціону, білків та жирів — 18-22% та 23-27% відповідно.

Для підбору раціону на добу, що задовольняє вимоги дієти, було взято з сучасних статей та досліджень [36-38], дивись табл. 3.15.

Таблиця 3.15

Приклад добового раціону DASH-дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Вівсяні пластівці	150,00	4,80	600,00	19,71	10,31	102,14
Знежирене молоко	250,00	9,60	77,50	7,50	0,25	11,75
Яблуко	176,00	3,01	110,88	0,65	0,70	22,88
Куряче філе	85,00	13,35	122,40	26,52	1,45	0,77
Варений бурий рис	100,00	5,70	111,00	2,60	0,80	24,20
Печена картопля	280,00	4,20	238,00	5,60	0,28	53,20
Індичка	90,00	11,16	121,50	20,88	3,60	0,27

Продовження таблиці 3.15

Помідори	40,00	4,12	8,00	0,36	0,08	1,56
Горіхи	20,00	17,00	143,60	3,15	12,68	3,69
Йогурту без цукру	250,00	22,25	267,50	10,00	20,00	11,75
Сумарні значення		77,11	1 800,38	21,55%	25,06%	51,59%

Для створення продуктового кошика було виконано моделювання дієти для жінки 54 років, зростом 172 та вагою 65 кг, що має середній рівень фізичної активності, займаючись фізичними вправами 2-3 рази на тиждень. Середня рекомендована кількість калорій на добу складає 2000 ккал. Згідно введених даних був сформований продуктивний кошик, дивись табл. 3.16.

Таблиця 3.16

Результат оптимізації продуктового кошика для DASH-дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Знежирене молоко	150,00	0,75	18,00	1,05	4,17	1,50
Горіхова каша	200,00	14,00	194,00	1,18	3,33	34,00
Пластівці вівсяні	200,00	26,28	800,00	13,74	6,40	136,18
Яйця курячі	1,00	0,12	1,60	0,10	5,98	0,01
Хліб житній	130,00	9,75	313,30	1,30	4,47	62,79
Капуста білокачанна	28,00	0,56	8,69	0,07	0,31	1,12
Буряк	14,00	0,29	6,50	0,01	0,15	1,44
Яблуко українське	100,00	0,37	63,00	0,40	1,71	13,00
Бекон	38,00	3,75	183,40	18,75	9,43	0,00
Курятина гомілка	159,00	33,41	189,30	5,41	13,83	0,00
Гречка	33,00	0,99	34,79	0,21	0,87	6,63
Олія соняшника	13,00	0,00	116,61	13,00	0,86	0,00
Сумарні значення		52,15	2006,00	25%	18%	55%

За даними таблиць можна побачити, що обидва раціону задовольняють співвідношення основних макронутрієнтів у повній мірі. За рахунок підбору більш дешевих продуктів результати розрахунку програми мають меншу вартість при більшій кількості енергії, але при цьому поступаються різноманіттю продуктів. Головним джерелом вуглеводів виступає вівсяна та горохова каша, а також житній хліб. Рекомендовані раціони включають більше різноманіття видів м'яса та риби, тоді як у результатах розрахунку це здебільшого м'ясо курки.

Додавання бекону та олії до раціону при цьому виді дієти є спірним та залежить від рівню обмежень дієти. У результатах розрахунків їх кількість не перевищує рекомендовану, тому вони можуть бути залишені як джерела жирів та енергії.

Рекомендації до вегетаріанської дієти не мають різниці порівняно з загальними рекомендаціями здорового харчування: 45-65% вуглеводів, 10-35% білків, 20-35% жирів на добу. За рахунок відмови від м'яса тваринного походження, джерелами білків виступають здебільшого каші, бобові та горіхи. У прикладі було взято раціони для лактоовоовегетаріанської дієти, яка дозволяє включення продуктів тваринного походження таких як: яйця, молочні продукти [39-40], дивись табл. 3.17.

Таблиця 3.17

Приклад добового раціону вегетаріанської дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Вівсянка	150,00	4,80	600,00	19,71	10,31	102,14
Яблуко	140,00	2,39	88,20	0,52	0,56	18,20
Банан	180,00	10,26	169,20	2,16	0,36	39,60
Картопля	200,00	3,00	170,00	4,00	0,20	38,00
Тофу	50,00	13,35	91,50	10,35	5,30	0,65
Капуста	120,00	1,44	37,20	2,40	0,31	4,80

Продовження таблиці 3.17

Горіхи	30,00	25,50	215,40	4,73	19,01	5,54
Авокадо	100,00	19,9	159,00	1,60	13,10	6,00
Оливкова олія	17,20	17,17	154,11	0,01	17,10	0,03
Сумарні значення		96,91	1 700,61	10,86%	35,14%	51,29%

Оптимізація кошику для вегетаріанської дієти були проведена для жінки, 25 років, 158 см зростом та вагою 58 кг з помірними тренуваннями 1-3 рази на день. Рекомендована кількість калорій на добу буде складами близько 1500 ккал. Максимальна дозволена вартість кошика була встановлена у 120 грн. Результати оптимізації дивись у табл. 3.18.

Таблиця 3.18

Результат оптимізації продуктового кошика для вегетаріанської дієти

Продукт	Кількість, гр	Вартість, грн	Калорійність, ккал	Білки, гр	Жири, гр	Вуглеводи, гр
Горіх волоський	28,00	201,04	4,41	17,75	19,83	5,17
Квасоля кольорова	50,00	63,50	4,35	0,25	4,30	11,40
Пластівці вівсяні	99,00	395,87	13,00	6,80	3,17	67,39
Капуста	100,00	31,00	2,00	0,26	1,12	4,00
Перець болгарський	10,00	2,70	0,10	0,00	1,38	0,50
Яблуко українське	100,00	63,00	0,37	0,40	1,71	13,00
Кефір 2,5%	200,00	98,00	6,00	4,00	9,04	8,00
Вермішель коротка	99,00	357,68	14,78	0,99	2,84	69,96
Олія рафінована	13,00	116,61	0,00	13,00	0,86	0,00
Цукор кристалічний	20,00	80,20	0,00	0,00	0,64	20,00
Сумарні значення		44,89	1409,60	28%	13%	57%

Можна відмітити, що багато вегетаріанських дієт включають тофу у якості заміни м'ясним продуктам та авокадо як джерело корисних жирів. Вартість цих продуктів значно впливає на вартість всього кошику, яка вдвічі перевищує вартість кошику за результатами алгоритму, але і надає більшу кількість енергії.

Обидва раціону задовольняють бажані показники співвідношення енергетичних речовин. Можна відмітити відсутність великого набору фруктів та овочів у рекомендаціях алгоритму оптимізації та більший фокус на основному раціоні з квасолі, пластівців та вермішелі — продуктів, що мають високі показники вуглеводів та білків.

За результатами аналізу та порівняння дієт можна зробити наступні висновки:

1. Результати роботи програмного продукту повноцінно задовольняють щоденно потребу енергії та балансу основних макронутрієнтів.
2. Результати розрахунків пропонуються такі комбінації продуктів, що ефективно зменшують загальну вартість кошику.
3. Завдяки роботі програмного продукту підвищується зручність перегляду, аналізу за харчового складу продуктів та контролю за щоденним раціоном.
4. Результати розрахунків поступаються у різноманітті та вмісті овочів та фруктів у раціоні порівняно з крупами, молочними та м'ясними продуктами.

3.6. Висновки

Було розроблено архітектуру та створено веб-додаток для оптимізації щоденного раціону харчування. Dodatok вкладається з веб-серверу, створеному за допомогою фреймворку Django, бази даних та СУБД PostgreSQL, клієнту, створеному за допомогою фреймворку Angular.

Dodatok включає авторизації, реєстрації, адміністрування та перевірки вхідних даних, виконання розрахунків та перегляду їх результатів.

Було розроблено та описано API запити серверу, а також механізм взаємодії між клієнтом та сервером.

Інтерфейс додатку налаштований на зручність користування з пристроїв з різним розміром екрану та операційною системою, з можливістю налаштування візуальної теми сторінок.

За введеними вхідними даними програмний додаток виконує моделювання та оптимізацію продуктового кошику та повертає психіко продуктів, енергетична цінність яких задовольняє потребам користувача, а баланс білків, жирів та вуглеводів знаходиться у рекомендованих межах.

Для перевірки доцільності впровадження програмної системи було проведено аналіз та порівняння результатів оптимізації з прикладами раціону відповідних дієт. За результатами перевірки можна зробити висновки, що додаток успішно розраховує продуктові кошики, що відповідають умовам дієти та знаходить такий набір продуктів, що дозволяє зменшити витрати на харчування без втрати якісної кількості калорій та основних макронутрієнтів. З недоліків результатів оптимізації можна визначити менше різноманіття продуктів та відсутність достатньої кількості продуктів з категорії овочі та фрукти, що може буде виправлено додаванням більш складних умов в розрахунки, які будуть брати до уваги окрім білків, жирів та вуглеводів, також баланс вітамінів та харчових волокон.

ВИСНОВКИ

Результатом виконання випускної кваліфікаційної роботи став програмний веб-додаток для оптимізації набору продуктів для збалансованого харчування.

Було проведено дослідження предметної області, рекомендацій щодо балансу макронутрієнтів дієт різного напрямку та щоденного споживання енергії. За результатами досліджень було побудовано та реалізовано обчислення математичної моделі з використанням бібліотеку Python SciPy.

Інтерфейс користувача було реалізовано за допомогою фреймворку Angular на мові TypeScript. Сервер для обробки запитів клієнта та виконання розрахунків було розроблено у фреймворці Django на мові програмування Python. Зберігання даних було організовано з допомогою СУБД PostgreSQL.

Додаток пропонує зручний інтерфейс для введення даних, що підлаштовується під розмір екрану користувача, має інтуїтивно зрозумілі елементи управління та дозволяє обирати бажаний стиль відображення сторінок.

Для збереження результатів розрахунків та персоніфікації профілю користувача додаток має систему реєстрації та авторизації, розроблені згідно сучасних заходів безпеки.

За результатами дослідження ефективності роботи додатку було зроблено наступні висновки:

- додаток є ефективним інструментом для розрахунку продуктового кошику, що має енергетичну цінність необхідного для задовільнення щоденних потреб користувача згідно його персональних характеристик та рівня фізичної активності та рекомендований обраною дієтою баланс макронутрієнтів;

- коректна конфігурація обмежень дозволяє моделювати продуктовий кошик, що задовольняє вимоги здорового балансу харчування та має мінімальну можливу вартість для заданого набору продуктів;

– так як додаток фокусується на балансі саме макронутрієнтів, тобто білків, жирів та вуглеводів, він потребує окремого налаштування для додавання в раціон більшої кількості овочів та фруктів. Для підвищення ефективності обчислень рекомендовано включити у процес розрахунку баланс також мікронутрієнтів, рекомендовані добові норми вітамінів та харчових волокон;

– так як метою є максимальне зменшення вартості продуктового кошика, результати розрахунків дозволяють ефективно планувати закупівлі при обмеженні бюджету, але пропонують менше різноманіття продуктів.

Напрямок подальшого розвитку програми оптимізації набору продуктів є додавання у розрахунки більше факторів, що визначають здорове різноманіття продуктів харчування.

По результатах виконання роботи було зроблено доклад на міжнародній науковій мультидисциплінарній інтернет-конференції «Світ наукових досліджень» №18 та отримано сертифікат [41].

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Hwalla N. Dietetic practice: the past, present and future / N. Hwalla, M. Koleilat. – East Mediterr : Health J, 2004. – 716 p.
2. Atwater W. O. The chemical composition and nutritive values of food-fishes and aquatic invertebrates / Wilbur Olin Atwater. – [S. l.] : Gov't print. off., 1891. – 214 p.
3. Мікро- та макронутрієнти - основні відмінності і правила використання [Електронний ресурс] // Здорове харчування. – Режим доступу: <https://sayyes.com.ua/ua/mikro-i-makronutrienty-osnovnye-otlichiya-i-pravila-ispolzovaniya/> (дата звернення: 17.09.2023). – Назва з екрана.
4. Швець О.В. Модель індивідуального збалансованого харчування у поєднанні з сучасною фармакотерапією як найбільш ефективна стратегія успішного лікування поширених гастроентерологічних захворювань. Гастроентерологія. Гепатологія. Колопроктологія. 2021. Т. 59, № 1. С. 21–23.
5. Рекомендації щодо харчування при високому артеріальному тиску [Електронний ресурс] // Кафедра внутрішньої медицини № 3 та ендокринології Харківського національного медичного університету. – Режим доступу: <http://vnmed3.kharkiv.ua/wp-content/uploads/2016/11/Dashpatient01.pdf> (дата звернення: 17.09.2023). – Назва з екрана.
6. Про затвердження Методичних рекомендацій для лікарів загальної практики – сімейної медицини з приводу консультування пацієнтів щодо основних засад здорового харчування [Електронний ресурс] : Наказ від 14.01.2013 р. № 16. – Режим доступу: https://zakononline.com.ua/documents/show/129815___129815 (дата звернення: 18.09.2023). – Назва з екрана.
7. Як визначити оптимальну вагу: формула індексу маси тіла [Електронний ресурс] // Міністерство охорони здоров'я України. – Режим доступу: <https://moz.gov.ua/article/health/jak-viznachti-optimalnu-vagu-formula-indeksu-masi-tila> (дата звернення: 15.10.2023). – Назва з екрана.

8. Здорове харчування: збірник матеріалів для працівників системи охорони здоров'я / укл.: В.В. Брич, В.Й. Білак-Лук'янчук, Г.О. Слабкий, І.Я. Гуцол, Н.Й. Потокій. - Ужгород, 2020. - 64 с.

9. Низьковуглеводна дієта - список продуктів і меню харчування на тиждень по днях - Преса Рівне [Електронний ресурс] // Преса Рівне. – Режим доступу: <https://pressa.rv.ua/news/nyzkovuglevodna-diyeta-spysok-produktiv-i-menuu-harchuvannya-na-tyzhden-po-dnyah/> (дата звернення: 16.09.2023). – Назва з екрана.

10. Здорове харчування - високобілкова дієта [Електронний ресурс] // «Живи Активно». – Режим доступу: <https://zhyvyaktyvno.org/news/sklki-blkupotrбно-organzmu> (дата звернення: 24.09.2023). – Назва з екрана.

11. High blood pressure and diet [Electronic resource] // National Library of Medicine. – Mode of access: <https://medlineplus.gov/ency/article/007483.html> (дата звернення: 19.10.2023). – Назва з екрана.

12. Кравчун Н.О. Сучасні підходи до лікування дисліпопротеїдемій у хворих на цукровий діабет: [методичні рекомендації] / Кравчун Н.О., Козаков О.В., Пилипенко І.І. – Київ-Харків: Інститут проблем ендокринної патології, 2005. – 24 с.

13. Калмикова Ю.С. Особливості лікувального харчування при цукровому діабеті. Педагогіка, психологія та медико-біологічні проблеми фізичного виховання і спорту. 2013. Т. 17, № 1. С. 30–31.

14. Швець О.В. Дієта при цукровому діабеті 2-го типу. Рекомендації для пацієнтів. Кафедра внутрішньої медицини №3 та ендокринології ХНМУ. URL: http://vnmed3.kharkiv.ua/wp-content/uploads/2014/05/diabetes2_diet.pdf (дата звернення: 23.09.2023). – Назва з екрана.

15. Castro-Quezada I. The mediterranean diet and nutritional adequacy: a review [Electronic resource] / Itandehui Castro-Quezada, Blanca Román-Viñas, Lluís Serra-Majem // Nutrients. – 2014. – Vol. 6, no. 1. – P. 231–248. – Mode of access: <https://doi.org/10.3390/nu6010231> (date of access: 27.09.2023). – Title from screen.

16. mindbodygreen [Electronic resource] // mindbodygreen: well-rounded well-being for a life well lived. – Mode of access: <https://www.mindbodygreen.com/articles/mediterranean-diet-macros> (дата звернення: 27.09.2023). – Назва з екрана.

17. Мінфін. Індекс цін на продукти [Електронний ресурс] / Мінфін // Ставки, індекси, тарифи. – Режим доступу: <https://index.minfin.com.ua/ua/markets/wares/index> (дата звернення: 20.11.2023). – Назва з екрана.

18. Яремчук Г. Скільки коштує продуктивний кошик українця у дрогобичі та які продукти у ньому є – дрогобич.city [Електронний ресурс] / Галина Яремчук // Дрогобич.City. – Режим доступу: <https://drohobych.city/articles/309708/skilki-koshtuye-produktivij-koshik-ukraincu-u-drohobichi-ta-yaki-produkti-u-nomu-ye> (дата звернення: 27.09.2023). – Назва з екрана.

19. Мінімальна заробітна плата (2000-2024) [Електронний ресурс] // Ставки, індекси, тарифи. – Режим доступу: <https://index.minfin.com.ua/ua/labour/salary/min/> (дата звернення: 15.11.2023). – Назва з екрана.

20. UN Healthy Diet & Weight Management Program | HR Portal [Electronic resource] // HR Portal. – Mode of access: <https://hr.un.org/page/healthy-weight-management-program> (дата звернення: 30.09.2023). – Назва з екрана.

21. Формула Миффлин-Сан-Жеора: як правильно розрахувати калорії і БЖУ для схуднення [Електронний ресурс] // Woman's world жіночий журнал. – Режим доступу: <https://w2w.com.ua/formyla-mifflina-san-jeora-dlia-rozrahynku-kalorii-dlia-cholovikiv-i-jinok/> (дата звернення: 14.12.2023). – Назва з екрана.

22. Non-Linear Least-Squares Minimization and Curve-Fitting for Python – Non-Linear Least-Squares Minimization and Curve-Fitting for Python [Електронний ресурс] // Site not found · GitHub Pages. – Режим доступу: <https://lmfit.github.io/lmfit-py/> (дата звернення: 14.12.2023). – Назва з екрана.

23. SciPy documentation – SciPy v1.11.4 Manual [Електронний ресурс] // Numpy and Scipy Documentation – Numpy and Scipy documentation. – Режим доступу: <https://docs.scipy.org/doc/scipy> (дата звернення: 14.12.2023). – Назва з екрана.

24. Hoppe P. D. Robert. H. Optimization Theory : навчальний посібник. New York City : HarperCollins Publishers, 2015. 56 р. [Електронний ресурс] // HarperCollins Publishers – Режим доступу: https://www.math.uh.edu/~rohop/fall_06/Chapter4.pdf (дата звернення: 10.10.2023).

25. Django overview [Електронний ресурс] // Django Project. – Режим доступу: <https://www.djangoproject.com/start/overview/> (дата звернення: 14.12.2023). – Назва з екрана.

26. bezkoder. Django: POST, PUT, GET, DELETE requests example | Rest Apis [Електронний ресурс] / bezkoder // BezKoder. – Режим доступу: <https://www.bzkoder.com/django-rest-api/> (дата звернення: 14.12.2023). – Назва з екрана.

27. Як використовувати JSON Web Tokens (JWT) для автентифікації [Електронний ресурс] // DevZone. – Режим доступу: <https://codeguida.com/post/1567> (дата звернення: 14.12.2023). – Назва з екрана.

28. JetBrains. PyCharm: the Python IDE for Professional Developers by JetBrains [Електронний ресурс] / JetBrains // JetBrains. – Режим доступу: <https://www.jetbrains.com/pycharm/> (дата звернення: 14.12.2023). – Назва з екрана.

29. Chacon S. Pro Git book / Chacon S., Straub B. – 2-ге вид. – New York : Apress, 2014. – 440 с.

30. Що таке Docker і навіщо він? [Електронний ресурс] // QualityAssuranceGroup. – Режим доступу: <https://qagroup.com.ua/publications/shcho-take-docker-i-navishcho-vin/> (дата звернення: 14.12.2023). – Назва з екрана.

31. What is Swagger [Електронний ресурс] // API Documentation & Design Tools for Teams | Swagger. – Режим доступу: <https://swagger.io/docs/specification/2-0/what-is-swagger/> (дата звернення: 14.12.2023). – Назва з екрана.

32. Tablycjakalorijnosti.com.ua - калорійність продуктів, ккал., калорії [Електронний ресурс] // Таблиця Калорійності. – Режим доступу: <https://www.tablycjakalorijnosti.com.ua/> (дата звернення: 14.12.2023). – Назва з екрана.

33. RD V. S. M. S. 1-Day High-Protein Meal Plan [Електронний ресурс] / Victoria Seaver M. S. RD // EatingWell. – Режим доступу: <https://www.eatingwell.com/article/290260/1-day-high-protein-meal-plan/> (дата звернення: 14.12.2023). – Назва з екрана.

34. Білкова дієта: плюси, мінуси та меню на тиждень [Електронний ресурс] // Новини України - останні новини України сьогодні - УНІАН. – Режим доступу: <https://www.unian.ua/common/10876577-bilkova-diyeta-plyusi-minusi-ta-menu-na-tizhden.html> (дата звернення: 14.12.2023). – Назва з екрана.

35. Білкова дієта: плюси, мінуси та меню на тиждень [Електронний ресурс] // Новини України - останні новини України сьогодні - УНІАН. – Режим доступу: <https://www.unian.ua/common/10876577-bilkova-diyeta-plyusi-minusi-ta-menu-na-tizhden.html> (дата звернення: 14.12.2023). – Назва з екрана.

36. Харчування і дієта при гіпертонії [Електронний ресурс] // IMPULS 24/7. – Режим доступу: <https://impuls24.com.ua/uk/harchuvannya-i-diyeta-pri-gipertonii/>. – Назва з екрана.

37. Поради лікаря. Дієта при артеріальній гіпертензії - МЦ "Країна Здоров'я" [Електронний ресурс] // МЦ "Країна Здоров'я". – Режим доступу: <https://kamenskoe.kraina-z.com.ua/aktsii-uk/poradi-likarya-diyeta-pri-arterialnij-gipertenzii/> (дата звернення: 14.12.2023). – Назва з екрана.

38. DASH-дієта при гіпертонії - шпиталь шептицького [Електронний ресурс] // Шпиталь Шептицького. – Режим доступу: <https://spital.org.ua/dashdiet> (дата звернення: 21.11.2023). – Назва з екрана.

39. Вегетаріанство: що це, користь та шкода, принципи меню - FitoBlog [Електронний ресурс] // Фітомаркет - Здоров'я, Краса та Спорт. – Режим доступу: <https://fitomarket.com.ua/ua/fitoblog/vegetarianstvo-racion-pitaniya-ili-stil-zhizni> (дата звернення: 22.11.2023). – Назва з екрана.

40. Вегетаріанське меню на тиждень: що можна їсти? ТОП-10 страв [Електронний ресурс] // Newfood-Media. – Режим доступу: <https://newfood.media/2022/01/17/vegetarianske-meniu-na-tyzhden-shcho-mozhna-isty-top-10-strav/> (дата звернення: 23.11.2023). – Назва з екрана.

41. Світ наукових досліджень. Випуск 18: матеріали Міжнародної мультидисциплінарної наукової інтернет-конференції, (м. Тернопіль, Україна – м. Переворськ, Польща, 20-21 квітня 2023 р.) / [редкол. : О. Патряк та ін.] ; ГО “Наукова спільнота”; WSSG w Przeworsku. – Тернопіль: ФО-П Шпак В.Б. – 286с.

КОД ПРОГРАМИ

У додатку приведено приклади коду класів сервісів, методів розрахунку, моделей та компонентів. У додатку наведено приклади 11 файлів, загальна кількість файлів – 238, вони додаються на оптичному носії разом з кваліфікаційною роботою.

Приклади ORM класів системи

ProductsBasket model

```
# клас для збегігання результатів оптимізації
class ProductsBasket(models.Model):
    #назва кошика
    name = models.CharField(max_length=300)
    # максимальна вартість
    max_sum = models.FloatField()
    #дата створення
    creation_date = models.DateTimeField(auto_now_add=True)
    # період створення
    period = models.IntegerField(validators=[MinValueValidator(0), MaxValueValidator(366)])
    # результати розрахунків
    products = models.JSONField()
    # користувач
    user = models.ForeignKey(User, on_delete=models.CASCADE)

    def __str__(self):
        return self.name
```

Diet model

```
from django.db import models

# клас для опису дієти
Diet(models.Model):
    name = models.CharField(max_length=250)
    description = models.CharField(max_length=1000)
    carbMin = models.DecimalField(max_digits=6, decimal_places=2)
    carbMax = models.DecimalField(max_digits=6, decimal_places=2)
    protMin = models.DecimalField(max_digits=6, decimal_places=2)
    protMax = models.DecimalField(max_digits=6, decimal_places=2)
    fatsMin = models.DecimalField(max_digits=6, decimal_places=2, default=0)
    fatsMax = models.DecimalField(max_digits=6, decimal_places=2)

    def __str__(self):
        return self.name
```

ShopProduct model

```
# модель для опису продукту у магазині
```

```

class ShopProduct(models.Model):
    # зв'язок з класом Продукт
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    # стани продукту
    states = models.ManyToManyField(ProductState)
    # назва
    name = models.CharField(max_length=350)
    # одиниця виміру
    unit = models.CharField(max_length=3)
    # кількість
    amount = models.DecimalField(max_digits=6, decimal_places=2)
    # вартість
    price = models.DecimalField(max_digits=6, decimal_places=2)
    # чи відноситься до базового набору
    default = models.BooleanField(default=True)

    def __str__(self):
        return self.name

```

Методи та класи обробки запитів та розрахунків

ProductCalcAPIView model

```

class ProductCalcAPIView(APIView):
    @swagger_auto_schema(operation_description="Get list of products baskets for user", responses={200:
ProductsBasketSerializer(many=True)})
    # 1. List all
    def get(self, request, *args, **kwargs):
        baskets = ProductsBasket.objects.filter(user_id=request.user.pk)
        serializer = ProductsBasketSerializer(baskets, many=True)
        return Response(serializer.data, status=status.HTTP_200_OK)

    @swagger_auto_schema(operation_description="Optimize products buckets for user",
responses={200: ProductsBasketSerializer()})
    def post(self, request, *args, **kwargs):
        products = request.data.get('products')
        diet_id = request.data.get('dietId')
        custom_energy_restrictions = request.data.get('energyRestrictions')
        energy_for_day = request.data.get('energyAmount')
        price_restriction = request.data.get('maxSum')
        term = int(request.data.get('term'))
        diet = Diet.objects.get(id=diet_id)
        energy_restrictions = custom_energy_restrictions if custom_energy_restrictions else
default_energy_restrictions
        diet_restrictions =
DietProductRestriction.objects.prefetch_related('restriction').filter(diet_id=diet_id).all();
        category_restrictions = DietCategoryRestriction.objects.filter(diet_id=diet_id).all()
        list_shop_products = products if len(products) else ShopProduct.objects.prefetch_related('states').all()
        products_list = prepare_products_for_calc(list_shop_products, len(products) == 0, diet_restrictions,
category_restrictions)
        optimization_results = calculate_basket_of_products(products_list, price_restriction, energy_for_day,
energy_restrictions, term)
        if request.user.is_authenticated:
            json_results = json.dumps(optimization_results, default=lambda o: o.__dict__, ensure_ascii=False,
sort_keys=True, indent=4)
            results = dict()
            results['user'] = request.user
            now = datetime.now()
            t_string = now.strftime("%d/%m/%Y %H:%M:%S")
            results['name'] = 'Список за ' + t_string
            results['period'] = term

```

```

    results['max_sum'] = price_restriction
    results['products'] = json_results

    ProductsBasket.objects.create(**results)
    return Response({'products': products_list, 'optimization': optimization_results},
status=status.HTTP_200_OK)

```

ProductCalcApiView model

```

from scipy.optimize import minimize

nutrients_energy = dict({
    'fats': 9,
    'proteins': 4,
    'carbohydrates': 4,
})

def get_constraints_fn(name, type, kkal, restrictions):
    def constraint_for_min(x: list, products):
        product = products
        energy_amount = 0
        for index, xi in enumerate(x):
            energy_amount = energy_amount + xi*product[index].get(name)
        return energy_amount*nutrients_energy.get(name) - kkal*restrictions.get(name)[0]

    def constraint_for_max(x: list, products):
        product = products
        energy_amount = 0
        for index, xi in enumerate(x):
            energy_amount = energy_amount + xi*product[index].get(name)
        return kkal*restrictions.get(name)[1] - energy_amount*nutrients_energy.get(name)
    return constraint_for_min if type == 'min' else constraint_for_max

def calculate_basket_of_products(all_products, sum_max_restriction, energy_per_day, restrictions, days =
1):
    energy_amount = energy_per_day * days

    # set function for objectives
    def objective(x: list):
        item = all_products
        price = 0
        for index, xi in enumerate(x):
            price = price + xi*item[index].get('price')
        return price

    def constraint_for_price(x: list, product_item):
        price = 0
        for index, xi in enumerate(x):
            price = price + xi * product_item[index].get('price')
        return sum_max_restriction - price

    def constraint_for_energy(x: list, product_item):
        sum_energy_amount = 0
        for index, xi in enumerate(x):
            sum_energy_amount = sum_energy_amount + xi * product_item[index].get('energy')
        return sum_energy_amount - sum_energy_amount

    def get_init_values(x):
        guesses = list()
        for index, xi in enumerate(x):
            guesses.append(1)

```

```

return guesses

# get min, max and equal bounds for calculations
def get_calculation_bounds():
    product_item = all_products
    bounds_list = list()
    for index, xi in enumerate(product_item):
        is_equal_bound = [restr for restr in xi.get('restrictions') if restr.get('comparator') == 'EQ']
        if len(is_equal_bound):
            bounds_list.append((is_equal_bound[0].get('amount')*days,
is_equal_bound[0].get('amount')*days))
            continue
        restriction_for_min_value = [restr for restr in xi.get('restrictions') if restr.get('comparator') == 'GT']
        restriction_for_max_value = [restr for restr in xi.get('restrictions') if restr.get('comparator') == 'LT']
        bound_min = restriction_for_min_value[0].get('amount') if len(restriction_for_min_value) else 0
        bound_max = restriction_for_max_value[0].get('amount') if len(restriction_for_max_value) else 200

        bound = (bound_min*days, bound_max*days)
        bounds_list.append(bound)
    return bounds_list

# set list of calculation constraints
price_constr = {'type': 'ineq', 'fun': constraint_for_price, 'args': [all_products]}
energy_const = {'type': 'eq', 'fun': constraint_for_energy, 'args': [all_products]}
proteins_const_min = {'type': 'ineq', 'fun': get_constraints_fn('proteins', 'min', energy_amount, restrictions),
    'args': [all_products]}
proteins_const_max = {'type': 'ineq', 'fun': get_constraints_fn('proteins', 'max', energy_amount, restrictions),
    'args': [all_products]}
fats_const_min = {'type': 'ineq', 'fun': get_constraints_fn('fats', 'min', energy_amount, restrictions),
    'args': [all_products]}
fats_const_max = {'type': 'ineq', 'fun': get_constraints_fn('fats', 'max', energy_amount, restrictions),
    'args': [all_products]}
carbohydrates_const_min = {'type': 'ineq',
    'fun': get_constraints_fn('carbohydrates', 'min', energy_amount, restrictions),
    'args': [all_products]}
carbohydrates_const_max = {'type': 'ineq',
    'fun': get_constraints_fn('carbohydrates', 'max', energy_amount, restrictions),
    'args': [all_products]}

constr_list = [price_constr,
    energy_const,
    carbohydrates_const_min,
    carbohydrates_const_max,
    fats_const_min,
    fats_const_max,
    proteins_const_min,
    proteins_const_max]

bounds = get_calculation_bounds()

# optimize products list
sol = minimize(objective, get_init_values(all_products), method='SLSQP', bounds=bounds,
constraints=constr_list)
results_list = sol.x

res = list()

# proces results
sum_values = {
    'price': 0,
    'energy': 0,
    'carbohydrates': 0,

```

```

        'proteins': 0,
        'fats': 0,
    }
    for index, product_item in enumerate(all_products):
        sum_values['energy'] = sum_values['energy'] + results_list[index] * product_item.get('energy')
        sum_values['price'] = sum_values['price'] + results_list[index] * product_item.get('price')
        sum_values['carbohydrates'] = sum_values['carbohydrates'] + results_list[index] *
product_item.get('carbohydrates')
        sum_values['proteins'] = sum_values['proteins'] + results_list[index] * product_item.get('proteins')
        sum_values['fats'] = sum_values['fats'] + results_list[index] * product_item.get('fats')

    if round(results_list[index], 4) != 0:
        product_res = {
            'name': product_item.get('name'),
            'id': product_item.get('id'),
            'price': round(results_list[index] * product_item.get('price'), 2),
            'unit': product_item.get('unit') if product_item.get('unit') == 'шт' else 'гр',
            'amount': round(results_list[index]),
            'states': [{
                'state': product_item.get('state'),
                'proteins': round(results_list[index] * product_item.get('proteins'), 2),
                'carbohydrates': round(results_list[index] * product_item.get('carbohydrates'), 2),
                'fats': round(results_list[index] * product_item.get('fats'), 2),
                'energy': round(results_list[index] * product_item.get('energy'), 2),
            }]
        }
        res.append(product_res)
    return {'product_bucket': res, 'general': sum_values}

```

Приклади складових клієнта

Diet Form Component TS file

```

@Component({
  selector: 'app-diet-select-block',
  templateUrl: './diet-select-block.component.html',
  styleUrls: ['./diet-select-block.component.scss']
})
export class DietSelectBlockComponent {

  @Input() dietForm: FormGroup;
  @Input() diets: IDiet[] = [];
  @Input() categories: ICategory[] = [];

  selectWorkType(dietType: number): void {
    this.dietForm.get('dietId')?.setValue(dietType);
    if (dietType === 7) {
      const categoriesToExclude = this.dietForm.get('categoriesToExclude')?.value as number[] || [];
      categoriesToExclude.push(3);
      this.dietForm.get('categoriesToExclude')?.setValue(categoriesToExclude);
    }
  }
}

```

Diet Form Component HTML file

```

<h3 class="headline">Оберіть тип дієти для розрахунку</h3>
<mat-accordion>
  <mat-expansion-panel *ngFor="let diet of diets" [ngClass]="{'selected': dietForm.value.dietId ===
diet.id}">

```

```

<mat-expansion-panel-header (click)="selectWorkType(diet.id)">
  <mat-panel-title>
    <span class="diet-title">
      {{ diet.name }}
    </span>
  </mat-panel-title>
</mat-expansion-panel-header>
<mat-accordion>
  {{ diet.description }}
</mat-accordion>
</mat-expansion-panel>
</mat-accordion>
<div class="selection-block" [formGroup]="dietForm">
  <h3 class="headline">Прибрати продукти з категорії:</h3>
  <mat-button-toggle-group formControlName="categoriesToExclude"
    class="exclude-list"
    appearance="legacy"
    multiple>
    <mat-button-toggle *ngFor="let category of categories" [value]="category.id">{{ category.name }}</mat-
button-toggle>
  </mat-button-toggle-group>
</div>
<div class="selection-block">
  <a mat-raised-button routerLink="/products" [queryParams]="{from: 'calculations'}">
    Преглянути список продуктів для розрахунків
  </a>
</div>

```

Calculations Service

```

export class UserCalculationsService {
  // calculate daily energy by the Mifflin-San Geor formula
  getDailyEnergy(inputValues: IUserParams): number {
    let energyAmount =
      (10 * inputValues.weight) + (6.25 * inputValues.height)
      - (5 * inputValues.years)
      + ENERGY_GENDER[inputValues.sex];

    const kkalPerDay = Math.round(this.calculateWithActivityLevel(energyAmount,
inputValues.activityLevel));
    return kkalPerDay;
  }
  // calculate daily energy with activity level
  private calculateWithActivityLevel(energy: number, level: ActivityLevel): number {
    return Activity[level] * energy;
  }

  // calculate body index mass
  getBMI(userData: IUserParams): number {
    return +(userData.weight / Math.pow(userData.height / 100, 2)).toFixed(2);
  }
}

```

Optimization API Service

```

export class OptimizationService {

  constructor(
    private http: HttpClient,
  ) {}
}

```

```

// Optimize Products List by user input
getOptimizationResults(body: ICalculationsState): Observable<IProductBasketResult> {
  return this.http.post<IProductBasketResult>(`${environment.apiUrl}products/optimization`, body);
}

// Get Product Basket by ID
getProductBasket(id: number): Observable<IProductBasket> {
  return this.http.get<any>(`${environment.apiUrl}products/product-basket/${id}`)
    .pipe(
      map(basket => ({
        id: basket.id,
        name: basket.name,
        period: basket.period,
        maxSum: basket['max_sum'],
        creationDate: basket['creation_date'],
        products: JSON.parse(basket.products)
      })))
    );
}

// Get current user Product Baskets
getProductsList(): Observable<IProductBasket[]> {
  return this.http.get<IProductBasketResponse[]>(`${environment.apiUrl}products/optimization`)
    .pipe(
      map(res => res.map(basket => ({
        id: basket.id,
        name: basket.name,
        period: basket.period,
        maxSum: basket.max_sum,
        creationDate: basket.creation_date,
        products: JSON.parse(basket.products)
      }))))
    );
}

// Get Product Baskets as File
getProductAsFile(id: number, downloadAsFile: string): Observable<any> {
  return this.http.get<any>(`${environment.apiUrl}products/product-
basket/${id}?downloadAsFile=${downloadAsFile}`, {
  responseType: 'blob' as "json"
});
}
}
}

```

Optimization Actions

```

export class InitBasketState {
  static readonly type = `[${stateNames.calculations}] init calculations state`;
}
export class SetBasketFormData {
  static readonly type = `[${stateNames.calculations}] set basket form data`;
  constructor(public payload: ICalculations) {}
}
export class SetUserCalcData {
  static readonly type = `[${stateNames.calculations}] set userC calc data`;
  constructor(public payload: IUserParams) {}
}
export class ResetFormData {
  static readonly type = `[${stateNames.calculations}] reset basket form data`;
}

```

ПЕРЕЛІК ФАЙЛІВ НА ДИСКУ

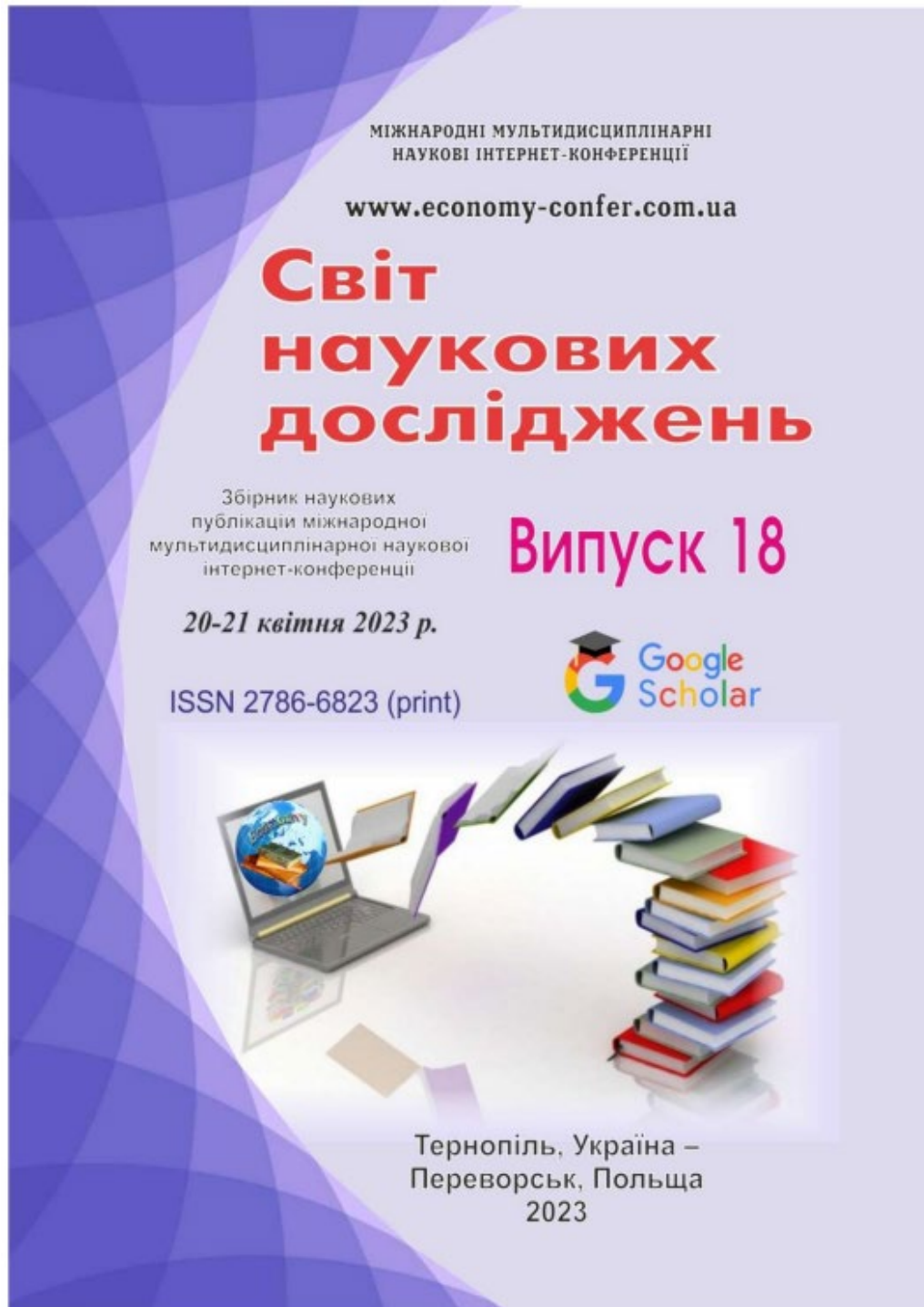
Ім'я файлу	Опис
Пояснювальні документи	
Кваліфікаційна робота_ Kurakina_A_S.docx	Пояснювальна записка до кваліфікаційної роботи. Документ Word.
Кваліфікаційна робота_ Kurakina_A_S.pdf	Пояснювальна записка до кваліфікаційної роботи в форматі PDF
Програма	
Kurakina_Products_server.rar	Архів. Містить коди програми серверу та її залежості
Kurakina_Products_client.rar	Архів. Містить коди програми клієнту та її залежості
Презентація	
Kurakina_A_S.ppt	Презентація кваліфікаційної роботи

ТЕЗИ ДОПОВІДІ НА МІЖДУНАРОДНІЙ КОНФЕРЕНЦІЇ

Тези Міжнародної мультидисциплінарної наукової інтернет-конференції

«Світ наукових досліджень» №18

Веб-додаток для моделювання дієти при лікуванні діабету другого типу



<i>Маркова Дар'я Олександрівна</i> ІТ-МАРКЕТИНГ, ЯК СПОСІБ РОЗВИТКУ УКРАЇНСЬКОЇ ТОРГІВЛІ У СФЕРІ МАРКЕТИНГУ.....	43
<i>Миколайчук Ірина Павлівна</i> ЕФЕКТИВНІСТЬ КОМАНДНОЇ МОДЕЛІ РОБОТИ ОРГАНІЗАЦІЇ.....	45
<i>Мірошник Марія Володимирівна, Шапран Олена Євгенівна</i> ЕВОЛЮЦІЯ КОНКУРЕНЦІЇ ЯК ЕЛЕМЕНТА РИНКОВОЇ ЕКОНОМІКИ.....	47
<i>Романькова Оксана Миколаївна, Висоцький Володимир Михайлович</i> БЮДЖЕТНИЙ ДЕФЦИТ УКРАЇНИ ТА ШЛЯХИ ЙОГО ПОДОЛАННЯ.....	48
<i>Тараненко Ганна Дмитрівна, Західна Оксана Романівна</i> РОЛЬ МІЖБЮДЖЕТНИХ ТРАНСФЕРТІВ У ФОРМУВАННІ РЕСУРСІВ ОРГАНІВ МІСЦЕВОГО САМОВРЯДУВАННЯ.....	52
Інформаційні системи і технології	
<i>Бердник Михайло Геннадійович, Куракіна Анастасія Станіславівна</i> ВЕБ-ДОДАТОК ДЛЯ МОДЕЛЮВАННЯ ДІЄТИ ПРИ ЛІКУВАННІ ДІАБЕТУ ДРУГОГО ТИПУ.....	56
<i>Ковальов Микола Олександрович</i> ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЦИФРОВИХ ІНТЕГРУЮЧИХ СТРУКТУР.....	59
<i>Кравчук Олександр Віталійович</i> РОЛЬ ДОЗВОЛІВ ANDROID У ВИЯВЛЕННІ НЕБЕЗПЕЧНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ З ВИКОРИСТАННЯМ СТАТИЧНОГО АНАЛІЗУ.....	64
<i>Хмелюк Альона Василівна, Черненко Поліна Володимирівна</i> ОБЛІКОВА ПОЛІТИКА: СПОСОБИ ФОРМУВАННЯ НАКАЗУ.....	69
<i>Штогрин Павло Петрович, Боярінова Юлія Євгенівна</i> ДОСЛІДЖЕННЯ ПОТЕНЦІАЛУ АВТОПЛОТА БІЛА НА ОСНОВІ МУРАШИНОГО АЛГОРИТМУ.....	74

Інформаційні системи і технології

ВЕБ-ДОДАТОК ДЛЯ МОДЕЛЮВАННЯ ДІЄТИ ПРИ ЛІКУВАННІ ДІАБЕТУ ДРУГОГО ТИПУ

Бердник Михайло Геннадійович

доктор технічних наук,

Національний технічний університет

«Дніпровська політехніка»

ORCID: 0000-0003-4894-8995

Куракіна Анастасія Станіславівна

студент, Національний технічний університет

«Дніпровська політехніка»

Інтернет-адреса публікації на сайті:

<http://www.economy-confer.com.ua/full-article/4455/>

Питання здоров'я та харчування завжди тісно пов'язані між собою. Від щоденного раціону, якості, кількості та складу продуктів залежить чи буде отримувати організм людини належну кількість елементів для повноцінного функціонування. Однією з багатьох хвороб, що потребує ретельного догляду за харчуванням та вибором продуктів для щоденного раціону, є цукровий діабет [1-4]. Дієта відіграє ключову роль у лікуванні цукрового діабету другого типу. Склад дієти включає підтримку балансу у споживанні вуглеводів, жирів та білків.

Досить часто стикаючись з необхідністю підтримки такого типу дієти, людини необхідно проводити окремі дослідження щодо складу та харчової цінності продуктів, вивчати які товари пропонують магазини та думати: як на певну виділену суму придбати усе потрібне для раціону.

Після дослідження та вивчення наявних рішень для розв'язання даного типу проблем, було виявлено нестачу програмних продуктів та калькуляторів для швидкого розрахунку рекомендованого списку продуктів з урахуванням індивідуальних характеристик користувача та пропозицій українського ринку.

В роботі представлений веб-додаток для моделювання дієти при цукровому діабеті другого типу.

Розроблений програмний додаток являє собою комбінацію веб-сервера та веб-клієнта, комунікація яких відбувається через протокол HTTP/S. Робота з інтерфейсом користувача включає три головних сценарії: робота з додатком без реєстрації, робота авторизованого користувача, робота адміністратора додатка.

Користувач має можливість введення наступних даних:

- дані для реєстрації (ім'я користувача, email, пароль);
- дані для створення та редагування розрахунку щоденної норми калорій (стать, вік, вага, зріст, рівень фізичної активності);

- дані для авторизації (ім'я користувача, пароль);
- дані для створення та редагування продукту (назва продукту, категорія, вартість, кількість, одиниця виміру, стани продукту);
- дані для створення та редагування обмеження (продукт, тип обмеження, кількість);
- дані для розрахунку продуктового кошика (термін розрахунку, максимальна вартість).

При введенні даних користувачем відбуваються перевірки на коректність вхідних даних: відповідність потрібному типу, максимальні, мінімальні значення, перевірка на відповідність формату. Аналогічні перевірки відбуваються на боці сервера перед збереженням даних у базі.

За введеними користувачем даними додаток виконує розрахунок оптимального набору продуктів, що задовольняють обмеженням та мають потрібний при діабеті другого типу баланс білків, жирів та вуглеводів.

Результати розрахунків розподілені на дві сторінки, перша з яких містить загальну інформацію щодо результатів розрахунків: загальну кількість калорій, вартість, баланс білків, жиру та вуглеводів (рис. 1).

Друга сторінка містить список продуктів з інформацією щодо їх енергетичної цінності та вартості. Також тут запропоновані можливості для друку сторінки або завантаження її у вигляді файлів різного формату (рис. 2).

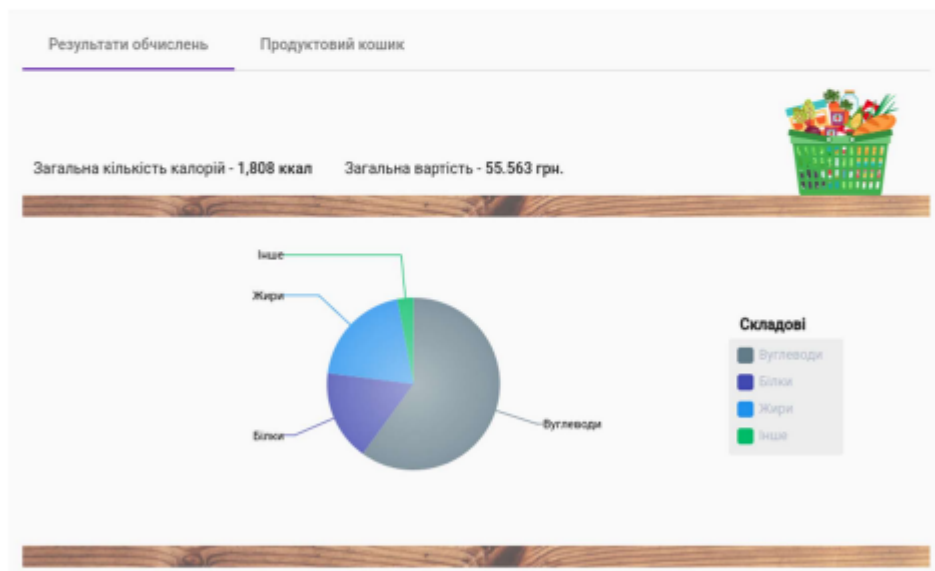


Рис. 1. Загальні результати оптимізації.

Для повноцінного функціонування додатка потрібна ЕОМ для розгортання серверів Django та Angular додатків. Сервера можуть знаходитися як на одному пристрої, так і на різних. Мінімальні технічні вимоги для запуску сервера: накопичувач [HDD]: 500 МБ; кількість ядер процесора: 4; оперативна пам'ять [RAM]: 512 МБ.

Таким чином веб-додаток надає можливість виконувати оптимізацію продуктового раціону шляхом розрахунку добової норми калорій згідно з індивідуальними характеристиками користувача, підбору оптимального співвідношення білків, жирів та вуглеводів у обраних продуктах та мінімізації вартості продуктового кошика.

Назва продукту	Кількість / Вага	Вартість	
Яйця перепелині (20 шт)	5 шт	11.02 грн.	
Батон Хлібодар Дніпровський пшеничний	96 гр	2.00 грн.	
Салат Пучок-Свіжачок	100 гр	23.47 грн.	
Перець болгарський жовтий	10 гр	1.34 грн.	
Курятина гомілка	118 гр	7.42 грн.	
Пшоно	107 гр	2.67 грн.	
Вермішель коротка	200 гр	5.68 грн.	
Олія Олейна рафінована	8 гр	0.55 грн.	

Рис. 2. Список продуктів раціону на добу.

Список використаних джерел:

1. What are the benefits of eating healthy. Medical News Today – *Articles*. URL: <https://www.medicalnewstoday.com/articles/322268> (дата звернення: 23.03.2023).
2. Type 2 diabetes. Mayo clinic – Patient Care & Health Information. URL: <https://www.mayoclinic.org/diseases-conditions/type-2-diabetes/symptoms-causes/syc-20351193> (дата звернення: 23.03.2023).
3. Швець О. В. Дієта при цукровому діабеті 2-го типу. Рекомендації для пацієнтів. Кафедра внутрішньої медицини №3 та ендокринології ХНМУ. URL: http://vnmed3.kharkiv.ua/wp-content/uploads/2014/05/diabetes2_diet.pdf (дата звернення: 24.03.2023).
4. Калмикова Ю. С. Особливості лікувального харчування при цукровому діабеті. Педагогіка, психологія та медико-біологічні проблеми фізичного виховання і спорту. 2013. Т. 17, № 1. С. 30-31.

СЕРТИФІКАТ УЧАСНИКА МІЖНАРОДНОЇ КОНФЕРЕНЦІЇ

